NAVAL RESEARCH LOGISTICS QUARTERLY. VOLUME 18, NUMBER 4

Office of Naval Research
Arlington, Virginia

December 1971

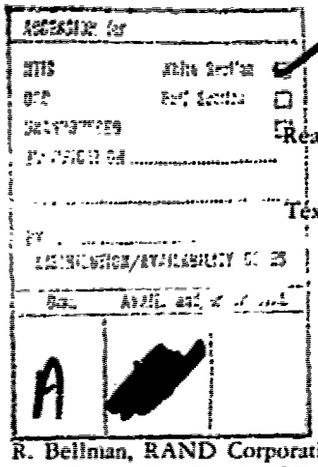
NAVAL RESEARCH LOGISTICS QUARTERLY

DECEMBER 1971
VOL. 18, NO. 4

OFFICE OF NAVAL RESEARCH

NAVSO P-1278

NAVAL RESEARCH LOGISTICS QUARTERLY



The Naval Research Logistics Quarterly is devoted to the dissemination of scientific information in logistics and will publish research and expository papers, including those in certain areas of mathematics, statistics, and economics, relevant to the over-all effort to improve the efficiency and effectiveness of logistics operations.

Information for Contributors is indicated on inside back cover.

The Naval Research Logistics Quarterly is published by the Office of Naval Research in the months of March, June, September, and December and can be purchased from the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. Subscription Price: $5.50 a year in the U.S. and Canada, $7.00 elsewhere. Cost of individual issues may be obtained from the Superintendent of Documents.

Issuance of this periodical approved in accordance with Department of the Navy Publications and Printing Regulations, NAVEXOS P-35

CONTROL VARIABLE METHODS IN THE SIMULATION OF A MODEL OF A MULTIPROGRAMMED COMPUTER SYSTEM

D. P. Gaver*

Naval Postgraduate School
Monterey, California

and

G. S. Shedler

IBM Research Laboratory
San Jose, California

ABSTRACT

One approach to the evaluation of the performance of multiprogrammed computer systems includes the development of Monte Carlo simulations of transitions of programs within such systems, and their strengthening by control variable and concomitant variable methods. An application of such a combination of analytical, numerical, and Monte Carlo approaches to a model of system overhead in a paging machine is presented.

I. INTRODUCTION

Many questions of interest which arise in the evaluation of the performance of multiprogrammed computer systems lead to stochastic (queuing) models that are too complex for existing mathematical techniques. One approach to the study of such models involves the development of Monte Carlo simulations, and their strengthening by control variable and concomitant variable methods, see [1]. The main idea is that control and concomitant techniques supplement and correct oversimplified, but, tractable analytical models. The goal is to obtain useful numerical results by means of which system performance can be judged.

This paper is concerned with the application of control variable methods to simulations of a model of a demand paging computer system. The particular objective is to obtain estimates of system overhead.

The computer which we consider in this paper is a single processor system with two-level memory which is multiprogrammed and operates in a demand paging environment. Such systems are described in [2, 5]. The following brief discussion gives the background necessary for an understanding of the models given in section III.

In a paging system all information that is explicitly addressable by the central processor (CPU) is divided into units of equal size called pages. The main memory (or execution store) is similarly divided into page-size sections called page-frames. In such machines it is possible to execute a program by supplying it with only a few page-frames of main memory. Having loaded the page containing the first executable instruction into a page-frame, execution begins and continues until an item of information required is not found in main memory. The page containing the missing information is then fetched, and overwrites a page currently in main memory; this is a continuing process. Thus in demand paging, information is brought into main memory only as a result of an attempt to use information

*This author is also a consultant to IBM Research.

not currently therein. An instance of this implicit "demand" for a page which is not in the store is termed a page exception. When dealing with large programs or in a multiprogramming mode in which the main memory is shared amongst several programs, it is usually the case that when another page has to be fetched from auxiliary memory the main store is filled. Consequently a choice must be made as to which page-frame in the main memory is to be overwritten. The rule governing this choice is called the replacement algorithm. It is usually the case that the content of the page-frame chosen to be overwritten must be transferred to auxiliary memory before the overwriting

The essential components of the hardware configuration which we consider are shown in Figure 1. The main memory M contains S page-frames. CCU is a channel control unit, and A is an auxiliary storage device. We assume that at all times $N(N \geq 2)$ problem programs $P_1, P_2, \ldots, P_N$ are being run in the system. A part of M is used as the residence of system (control) programs. Of the remaining S' page-frames of main memory, s_i page-frames are allocated to problem program P_i. Clearly we want

(1) $$\sum_{i=1}^{N} s_i = S', \text{ and if } l_i \text{ is the number of pages in program } P_i$$

the case of interest is that

(2) $$1 \leq s_i < l_i \quad \text{for} \quad 1 \leq i \leq N.$$

Under the multiprogramming assumption there is more than one program resident in the main memory ($N \geq 2$). There is, thus, contention for processing resources. Hence a conceptual queue is formed for processing services to be provided by the central processor unit (CPU). Whenever a program which is receiving processing service from the CPU references a page which is not in main memory, a request for data transfer service is made by the CPU to a channel control unit. The referenced page is then moved from auxiliary memory to main memory. Having initiated this request the CPU is free to render service to the next available program. A data transfer service consists of activity of a channel control unit and an input-output device (say a drum or a disk). Since we have assumed multiprogramming, there will sometimes be at least one request waiting for the service of data transfer. A second conceptual queue is formed for data transfer services to be provided by the data transfer unit (DTU). As soon as a referenced page is moved from auxiliary memory to main memory, the requesting program (logically) is again available for processing. It is assumed that the CPU can be operated concurrently with the DTU. Thus in the multiprogramming mode, the CPU can process one program while the DTU is processing page requests for other programs.

The foregoing discussion has made no mention of system overhead, that is, the processing done by the CPU to accomplish the switching from one problem to another, the construction and execution of appropriate channel control programs, and other activity, such as that required for queue management, and the execution of the replacement algorithm. In this paper a model is presented in which system overhead functions are represented explicitly. The results of the analysis provide a quantitative assessment of system overhead in terms of parameters which describe the demand upon the CPU for overhead activity, demand for processing, and also the paging characteristics of the program load. Details of this model and of a simplified model used as a control are given in the next section.

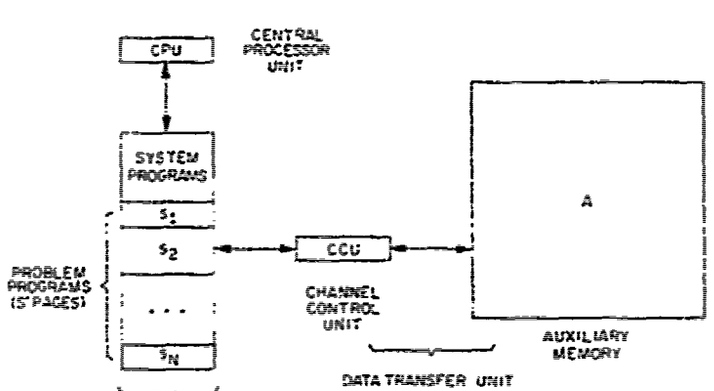
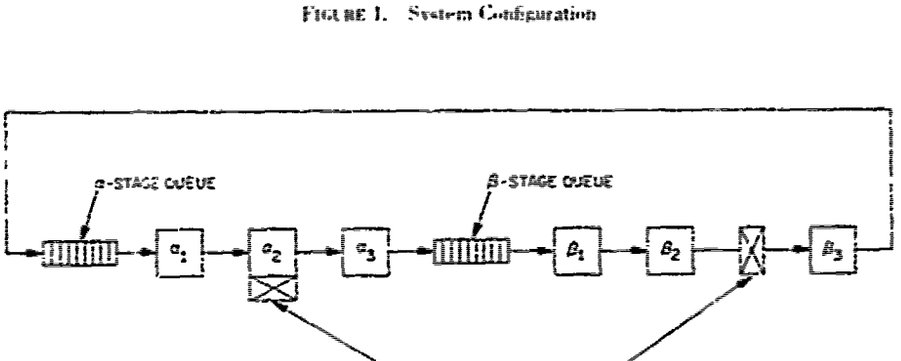


FIGURE 1. System Configuration



a CPU SERVICES $\alpha_1, \alpha_2, \alpha_3$, β_1 AND β_3.
b DTU SERVICES β_2
c NO β_1, β_3, SERVICE BY CPU DURING β_2 ACTIVITY OF THE DTU
d. $\alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3$ NON-INTERRUPTABLE
e. α_2 SERVICE HAS PRE-EMPTIVE RESUME TYPE INTERRUPTION AT COMPLETION OF β_2 SERVICE.

FIGURE 2. System Overhead Model

II. STRUCTURE OF THE MODELS

The model we consider consists of two sequential stages, the α-stage and the β-stage, see Figure 2. The system serves a constant number, N, of programs, $P_1, P_2, \ldots, P_N$ ($N \geq 2$), each of which goes through both stages in sequence and then returns to the first stage, this process being repeated continuously. Within the α-stage, a program receives, in order, each of the three services α_1, α_2, and α_3, and similarly, within the β-stage, a program receives each of three services, β_1, β_2, and β_3.

An interpretation of the six services α_1, α_2, α_3, β_1, β_2, and β_3 in this multiprogrammed demand paging system is as follows. Problem program processing corresponds to α_2 service and data transfer service (paging) corresponds to β_2 service. The remaining services α_1, α_3, β_1, and β_3 are interpreted as system overhead functions. Specifically we think of α_1 service as picking up the next program for

processing from the queue and restoring its machine state. We associate with α_3 the activity required to:

(i) save the machine state of the program relinquishing the CPU,

(ii) execute the replacement algorithm,

(iii) construct the channel control program for the required page, and

(iv) place an entry onto the paging queue.

The interpretation of β_1 is the activity required to

(i) pick up the next page request,

(ii) Start execution of the channel control program.

The placement of a new entry on the CPU queue and the termination of the input-output operation is associated with β_3 service. Under this interpretation, the major overhead activity is the α_3 service.

Six services are provided by but two servers, a single CPU and a single DTU. A β_2 service can be provided only by the DTU, while the remaining services are performed by the CPU. It is assumed that the CPU and the DTU can provide service simultaneously, subject to the restriction that no β_1 or β_2 service can be rendered by the CPU while the DTU is rendering a β_2 service. It is assumed that after having received α_3 service a program "moves" instantaneously from the α_3 service station to the tail of the queue in the β-stage, and after having received the β_3 service, "moves" instantaneously from the β_3 service station to the tail of the queue in the α-stage.

The single CPU renders α_1, α_2, α_3, β_1, and β_3 service to the several programs in the system. Having begun an α_1, α_2, β_1, or β_3 service, the CPU completes that service without interruption; however, an interruption of α_2 service occurs at any epoch at which a β_2 service is completed by the DTU. The interrupted α_2 service will be continued (after some time) from the point of interruption. Thus the "β-complete" interruption of an α_2 service is of the pre-emptive resume type.

At an epoch of completion of an α_1, α_2, α_3, β_1, or β_3 service and at an epoch of interruption of an α_2 service, i.e., at completion of β_2 service, the CPU chooses the next service to be rendered according to the following priority rule.

Rule of Priority Service:

(i) If there is a program waiting for β_2 service, begin that service.

(ii) If there is a program waiting for β_1 service, begin that service if DTU β_2 service is not in progress.

(iii) If the last α service rendered was a completed α_2 service, begin an α_3 service.

(iv) If the last α service rendered was an interpreted α_2 service, resume the α_2 service.

(v) If the last α service rendered was an α_1 service, begin an α_2 service.

(vi) If the last α service rendered was an α_3 service, and the queue at the beginning of the α-stage is not empty, begin an α_1 service.

If no claim is made on the CPU according to the rule of priority, the CPU is assumed to remain idle until the completion of the next β_2 service when the rule of priority is invoked. A flowchart for this CPU priority rule is given in Figure 3.

We assume that both the queue in the α-stage and the queue in the β-stage are served under a FIFO (First-In First-Out) queuing discipline.

Since no interruption of an α_1 or α_3 service can occur, a program completing a β_2 service while an α_1 or α_3 service is in progress must wait (in the β-stage) until the completion of that service before receiving β_3 service. Similarly a program whose α_2 service is interrupted by a β_2 completion must wait (in the α-stage) until a β_3 service and possibly a β_1 service has been rendered before its α_2 service is resumed. In fact if there is anyone in the β-stage queue the α_2 service is not resumed until β_2 service

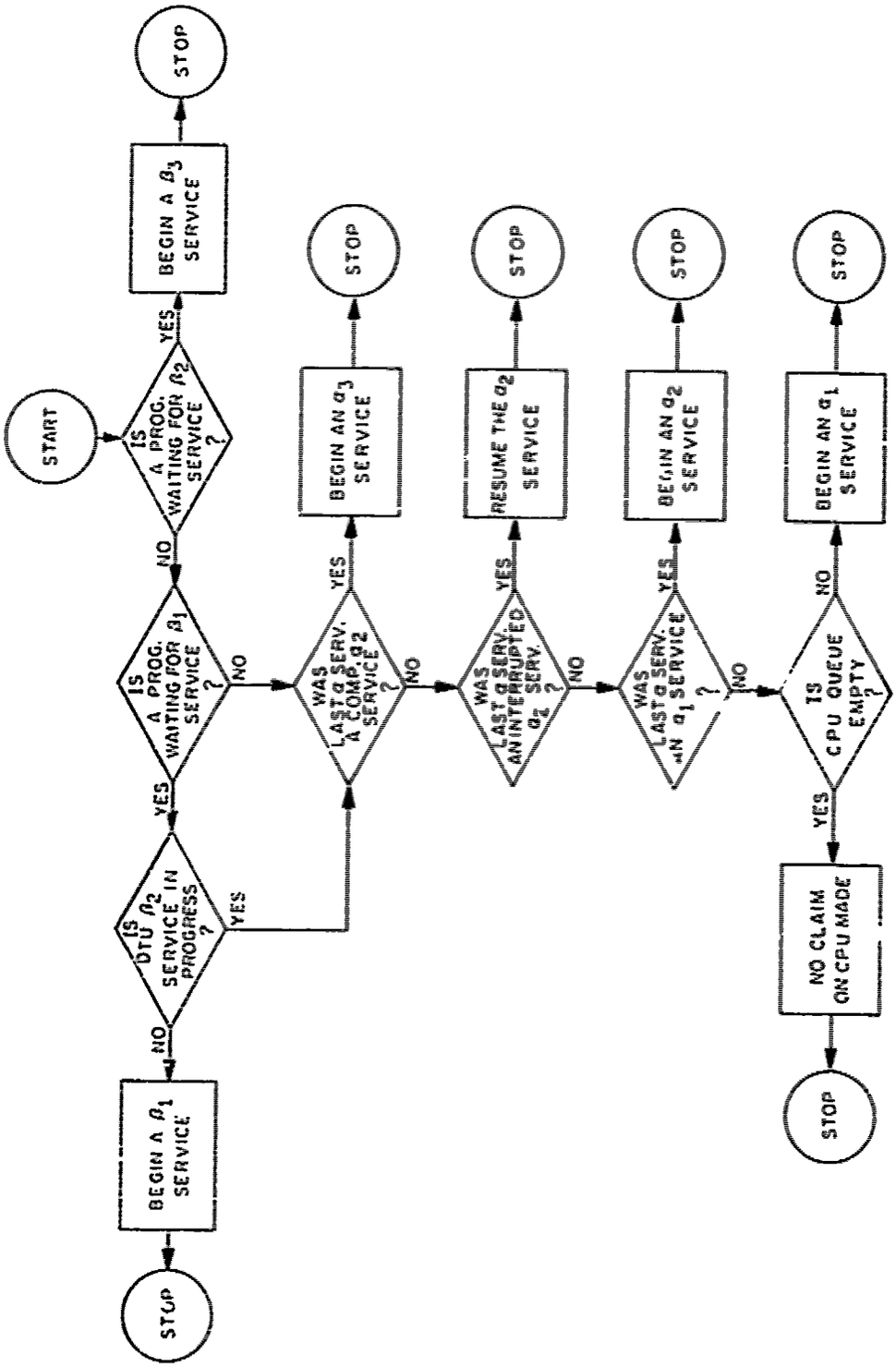


Figure 3. Rule of Priority Service

begins again (the DTU is put to work). Clearly, these priority rules are based on the assumption that the bottleneck in the multiprogrammed system is in fetching pages from auxiliary memory. This is generally true in present systems.

The Probabilistic Assumptions:

In the experiments conducted in this paper the following assumptions have been made.

(i) The successive α_1 and the successive α_3 service times are assumed to be independently and identically distributed positive random variables A_1 and A_3 with arbitrary distributions $F_{A_1}(t)$ and $F_{A_3}(t)$, e.g.,

$$F_{A_1}(t) = P\{A_1 \leq t\}.$$

(ii) For $l = 1, 2,$ and 3, the successive β_l service times ($l = 1, 2, 3$) are assumed to be independently and identically distributed random variables B_l ($l = 1, 2, 3$) with arbitrary distributions $F_{B_l}(t)$.

(iii) The successive α_2 service times are assumed to be independently and identically distributed random variables A_2 with exponential distribution having rate parameter λ_2, i.e.,

$$F_{A_2}(t) = P\{A_2 \leq t\} = 1 - e^{-\lambda_2 t} \quad (t \geq 0)$$

A mathematical analysis of this model has been given by Lewis and Shedler [3]. The result of the analysis presented in [3] using the above model is the determination of the long run fraction of time that each of the six services is in process, and hence the long run fraction of time that each of the two servers is busy.

Although assumption (iii) seems essential for an analytical treatment of the problem, no such simplification is necessary when simulations are being carried out. Indeed, one of the reasons for simulation arises from a desire to utilize other input processes.

Some remarks about the assumptions concerning α_2 service time are in order. We have assumed that each of the programs $P_1, P_2, \ldots, P_N$ is constrained to run in a memory smaller than its length, i.e., for all i, $s_i < l_i$. Under the demand paging assumption, a page is moved from the auxiliary memory to main memory only when it is needed and not already in main memory. Whenever P_i references a missing page while the portion of main memory allocated to P_i is filled to its capacity, a page in s_i is replaced by the newly referenced page, in accordance with the replacement algorithm. If the page to be replaced can only be one of the s_i pages in main memory belonging to program P_i, the replacement algorithm is said to operate locally. If the replacement algorithm is applied to the entire S' area of the main memory, it is said to operate globally. We consider an execution interval of a program to be a time interval during which the CPU can continue to process the program without referencing a page not contained in main memory. Thus program P_i, under a replacement algorithm which operates locally in a region of size s_i, gives rise to a sequence of execution intervals independent of the other programs. Although the length of an execution interval of program P_i is independent of the length of an execution interval of program P_j, $j \neq i$, under a replacement algorithm that operates locally, successive execution intervals of a single program might well not be statistically independent. We assume, however, that the combined sequence of execution intervals of the set of N programs which comprises the program load is independent, i.e., that successive execution intervals are independent.

Further, for the given program load, under a given memory partition $\pi(s_1, \ldots, s_n)$ and a specified replacement algorithm which operates locally, we assume that successive execution intervals are identically distributed, the distribution being exponential. It should be emphasized that the parameter of the exponential distribution is a function of the s_i which comprise the memory partition π. The appropriateness of the independence and exponential assumptions have been discussed by Smith [7] his arguments being supported by some empirical data.

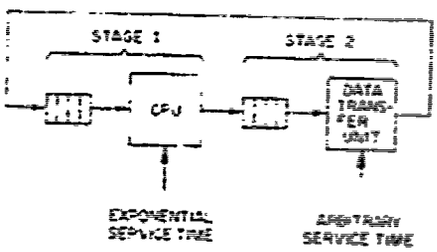


FIGURE 4 Control Model

III. THE CONTROL MODEL

The control model (see Figure 4) consists of two sequential stages, each stage acting as a single server. The system serves a constant number N of customers ($N \geq 2$), each of whom goes through both stages in sequence and then returns to the first stage; this process is repeated continuously. It is assumed that after completion of CPU service a program segment moves instantaneously from stage 1 to the tail of the queue in stage 2, and, after DTU service at that stage, back to the tail of the queue in stage 1.

An analysis of this model is given by Shedler [6] under the following assumptions.

Probabilistic Assumptions:

(i) The successive DTU service times are assumed to be independently and identically distributed as a random variable T with arbitrary distribution $F_T(t)$, i.e.,

$$F_T(t) = P\{T \leq t\}.$$

(ii) The successive CPU service times are assumed to be independently and identically distributed as a random variable X with exponential distribution having rate parameter λ, i.e.,

$$F_X(t) = P\{X \leq t\} = 1 - e^{-\lambda t} \quad (t \geq 0).$$

We state as a theorem two results from [6] which we shall use in the application of control variable methods to the simulation of the overhead model. The following definition introduces some notation.

Definition:

For $r = 1, 2, \ldots, N-1$ let

$$f_r(x)=\frac{\lambda}{(r-1)!}(\lambda x)^{r-1}e^{-\lambda x},\quad F_r(x)=1-\sum_{k=0}^{r-1}e^{-\lambda x}\frac{(\lambda x)^k}{k!}$$

and $R_r(x)=1-F_r(x)$ denote, respectively, the density function, the distribution function, and the survivor function of the gamma distribution with parameters λ and r. Let $F_0(t)\equiv 1$, $F_1(t)\equiv 0$ and

$$G_k=\int_0^\infty \left(F_k(t)-F_{k-1}(t)\right)dF_T(t)\quad (k=0,1,\ldots,N-2).$$

Then let C_i $(i=0,1,\ldots,N-1)$ be defined by

$$\begin{aligned}
&C_0=1\\
&C_1=C_0G_0\\
&C_2=((1-G_1)C_1-C_0)/G_0\\
&C_3=((1-G_1)C_2-G_2C_1)/G_0\\
&C_i=((1-G_1)C_{i-1}-G_2C_{i-2}-\cdots-G_{i-2}C_2-G_{i-1}C_1)/G_0\\
&\text{for } i=4,5,\ldots,N-1.
\end{aligned}$$

THEOREM: Let $A_1(t)$ (resp. $A_2(t)$) be the total amount of time that the CPU (resp. DTU) is busy during the interval $(0,t)$. Then

$$\lim_{t\to\infty}\frac{E(A_1(t))}{t}=\frac{(1-p_0)\frac{1}{\lambda}}{p_0\frac{1}{\lambda}+(1-p_0)E[T]}$$

$$\lim_{t\to\infty}\frac{E(A_2(t))}{t}=\frac{(1-p_0)E[T]}{p_0\frac{1}{\lambda}+(1-p_0)E[T]}$$

where $p_0=\dfrac{1}{\sum_{i=0}^{N-1}C_i}$ and the C_i are defined above.

IV. APPLICATION OF CONTROL VARIABLE METHODS TO MONTE CARLO SIMULATIONS

In multiprogrammed computer systems of the type considered in this paper, certain response variables are of particular interest. Some examples are the fraction of time spent by the CPU doing program processing (as opposed to processing associated with overhead functions or being idle) fraction of time DTU is busy, lengths of queues, waiting times in queues, etc. These response variables are influenced by other variables such as the degree of multiprogramming, the nature of the program load, characteristics of the physical devices, control strategy, and the like. In general, then, a response variable W is influenced by a set of system input variables, say $X_1, X_2, X_3, \ldots$ which we shall denote collectively by X. The probabilistic model provides a means of relating W to X, X being taken as a collection of random variables. Thus, the model gives rise to a known, but very complicated function

I such that

$$W = f(X). \tag{1}$$

and we seek information about characteristics such as the expected value $E[W]$ or the probability distribution of W. Simulation is a method for studying the distribution of W in which one observes sample values of W. We shall briefly outline the several simulation methods with which the paper is concerned.

A sample value of W is computed from (1), a sample value of X having first been obtained. A sample value of $X = (X_1, X_2, \ldots)$ can be found by first choosing a vector of pseudo random numbers uniformly distributed on the interval (0, 1) and converting these to realizations or samples of $X_1, X_2, \ldots$ perhaps by means of the probability integral transformation, i.e., from

$$X = F_1^{-1}(R), \text{ where} \tag{2}$$

$F_1(\cdot)$ is the probability distribution function of X and R is a random number uniformly distributed on (0, 1). In straightforward sampling, n independent realizations of W, denoted by $W_1, W_2, \ldots, W_n$, are obtained and averaged to give $\bar{W}$, an unbiased estimator of $E[W]$, i.e.,

$$\bar{W} = \frac{1}{n}\sum_{j=1}^{n} W_j. \tag{3}$$

The variance of the estimator $\bar{W}$ is $\frac{1}{n}\mathrm{Var}[W]$ and thus it is clear that the estimate can be brought closer to $E[W]$ by increasing n. Since, in general, rather long simulation studies will be required to represent adequately the behavior of queuing systems, the investigation of alternative techniques to straightforward sampling is of interest.

One such technique is antithetic variables, proposed for queuing problems by Page [4] and discussed in [1]. If a sample X results from a pseudo random number, R, and X is relatively large, then the sample X' resulting from $1-R$ will be relatively small. The antithetic idea is to create companion realizations, $W_j^{(1)}$ and $W_j^{(2)}$, resulting from antithetic realizations $X_j^{(1)}$ and $X_j^{(2)}$ in turn the result of R and $1-R$. The two antithetic realizations $W_j^{(1)}$ and $W_j^{(2)}$ are then averaged to obtain the estimate W_j. The average of $W_1, W_2, \ldots, W_n$ is taken as an estimator of $E[W]$, i.e.,

$$\widehat{E[W]}_a = \frac{1}{n}\sum_{j=1}^{n} W_j = \frac{1}{n}\sum_{j=1}^{n} \frac{(W_j^{(1)} + W_j^{(2)})}{2}. \tag{4}$$

The technique of antithetic variables is useful for reducing the sampling variability of simulations, but does not employ any information that may be known concerning the approximate behavior of the system under study. The control variable methods with which this paper is concerned involves the simultaneous use of simulation with approximate models. A simple estimating procedure that involves the use of an approximate model is as follows. We desire to estimate $E[W]$, where W is related to λ

by (1). We select an approximate model for which it is possible to calculate (analytically or numerically) the expectation of W^*, $E[W^*]$, relatively easily, W^* being the response variable of the model approximating that giving W. Although it is in practice likely that the distributions of W and W^* will be similar, the basic requirement is only that W and W^* be well correlated. Having chosen a control, then simulate W and W^* using the same random numbers R. That is, the input values X are identical across realizations to as great a degree as possible. This implies that W and W^* will be correlated. We then may estimate $E[W]$ as follows. In the present case, $E[W^*]$ is obtained numerically from the theorem of section III.

$$\widehat{E[W]}_c = E[W^*] + \frac{1}{n}\sum_{j=1}^{n} W_j - \frac{1}{n}\sum_{j=1}^{n} W_j^* = E[W^*] - \bar{W} - \bar{W}^*. \tag{5}$$

It is easily verified that the estimate (5) is unbiased. Further, we have

$$\mathrm{Var}\{\widehat{E[W]}_c\} = \frac{1}{n}\{\mathrm{Var}[W] + \mathrm{Var}[W^*] - 2\mathrm{cov}[W, W^*]\}. \tag{6}$$

so that an improvement over straightforward simulation has been obtained if W^* has the property that

$$\frac{\mathrm{cov}[W, W^*]}{\mathrm{var}[W^*]} > \frac{1}{2}. \tag{7}$$

In Table 1 we give results of experiments for straightforward sampling, in terms of which the other methods can be assessed. We display an estimate V of $\mathrm{Var}[\bar{W}]$ obtained from a set of $m = 20$ independent observations of $\bar{W}$ along with M the mean of the m observations. In one case the response

TABLE 1. Assessment of Straightforward Sampling

λ_2	CPU Utilization		DTU Utilization	
	M	V	M	V
2.00	65.662	1.85084	91.238	0.05175
1.00	91.090	1.00514	78.873	1.75506
0.50	99.765	0.01192	45.802	0.07449

variable W is CPU utilization and in the other case the response variable W is DTU utilization. In both cases, the system input variable X is the exponentially distributed α_2 service time. For positive integral c, CPU utilization $U_1(c)$ is defined by

$$U_1(c) = \frac{A_1(t_c)}{t_c} \times 100.$$

where $A_1(t_c)$ is the total amount of time that the CPU renders service (α_1, α_2, α_3, β_2, or β_3) the time interval $(0, t_c)$, and t_c is the epoch of simulated time at which the $(c+1)$th customer begins his α_1 service. DTU utilization $U_2(c)$ is defined similarly by

$$U_2(c) = \frac{A_2(t_c)}{t_c} \times 100,$$

where $A_2(t_c)$ is the total amount of time that the DTU renders service (β_2) during the time interval $(0, t_c)$.

All results displayed in this paper are for the case $c = 50$, are based on $n = 25$ independent realizations of W_1, and are for the case in which the degree of multiprogramming N is 4. All service distributions other than that of the α_2 service are constant. Unit time is taken to be the duration of a β_2 service. The duration of the dominant overhead service α_1 is 0.1, and the duration of an α_3, β_3, or β_1 is 0.02. In each realization, all customers are in the CPU queue at time $t = 0$. We observed that for a given c, termination of each realization when the $(c+1)$th customer was about to start his α_1 service yielded values of CPU and DTU utilization which differed only slightly from those obtained by the definition of a time interval $(0, t_c)$ by the first realization, and the termination of all subsequent realizations at simulated time t_c.

Results of experiments on the straight control method are given in Tables 2, 3, and 4. In Tables 2 and 3 we display results obtained from a single application of the method for CPU utilization and DTU utilization, respectively. We display in Table 4, both for CPU utilization and DTU utilization, an estimate V of $\mathrm{Var}\{\widehat{E[\Pi]}_c\}$ obtained from a set of $m = 20$ independent observations of $\widehat{E[\Pi]}_c$, along with M, the mean of the m observations. Comparison of these estimates of $\mathrm{Var}\{\widehat{E[\Pi]}_c\}$ with the estimates of $\mathrm{Var}\{\bar{\Pi}\}$ given in Table 1 indicates the reduction in variance obtainable by the straight control method.

TABLE 2. Straight Control Estimates CPU Utilization

λ_2	$E[\Pi^*]$	$\bar{\Pi}^*$	$\bar{\Pi}$	$\widehat{E[\Pi]}_c$
2.00	49,900	52,130	65,523	63,293
1.00	87,000	88,379	91,000	92,711
0.50	99,200	99,381	99,778	99,597

TABLE 3. Straight Control Estimates DTU Utilization

λ_2	$E[\Pi^*]$	$\bar{\Pi}^*$	$\bar{\Pi}$	$\widehat{E[\Pi]}_c$
2.00	99,900	98,735	94,282	95,148
1.00	87,000	85,807	79,019	80,212
0.50	49,600	49,283	45,305	46,122

TABLE 4. Assessment of Straight Control

λ_2	CPU Utilization		DTU Utilization	
	M	V	M	V
2.00	63.279	0.01497	95.460	0.00309
1.00	93.007	0.28159	80.609	0.06759
0.50	99.645	0.01537	46.155	0.03678

$$E[W]_c^0 = \bar{W}_i - (\bar{W}_i^* - E[W^*])$$

ith straight control estimate ($i=1, 2, \ldots, m=20$)

$$M = \frac{\sum_{i=1}^{20} E[W]_c^0}{20}$$

$$V = \frac{\sum_{i=1}^{20} (E[W]_c^0 - M)^2}{19}$$

The form of (7) suggests another possibility for improving precision, namely, that of a correction of the form

$$\widehat{E[W]}_r = \bar{W} + \beta(\bar{W}^* - E[W^*]), \tag{8}$$

where β is selected to minimize the variance of the estimate $E[W]_r$. If the optimum β,

$$\beta_0 = -\frac{\text{cov}[W, W^*]}{\text{Var}[W^*]}$$

is used, the resulting optimal regression adjusted control estimate has variance

$$\text{Var}\{\widehat{E[W]}_{r,0}\} = \frac{1}{n}\text{Var}[W]\{1 - (\text{corr}[W, W^*])^2\} \tag{9}$$

and therefore will, in theory, always be an improvement over simple estimates. Although Var[W^*] is presumably known, the required covariance will not be known and must be estimated from data. The realistic estimate uses an estimated optimum $\hat{\beta}_n$ and is of the form

$$\widehat{E[W]}_{r,0} = \bar{W} + \hat{\beta}_n(\bar{W}^* - E[W^*]), \tag{10}$$

where

$$\hat{\beta}_0 = -\frac{1}{n}\sum_{j=1}^{n}\frac{(W_j-\bar{W})(W_j^*-E[W^*])}{\mathrm{Var}[W^*]}.$$

It should be noted that the realistic estimate (10) may not be unbiased although the bias decreases as the sample size n increases.

Results of experiments on this control and regression method are given in Tables 5, 6, and 7. We again use the simple cyclic queue model (Figure 4) as a control for the overhead model (Figure 2). We then compute a regression adjusted estimate of the form (10), where

TABLE 5. Regression Adjusted Estimates CPU Utilization

λ_2	$E[W^*]$	β_0	$\bar{W}^*$	$\bar{W}$	$\widehat{E[W]}_r$
2.00	49.900	−0.9291	52.130	65.523	63.451
1.00	87.000	−0.6470	88.379	94.090	93.198
0.50	99.200	−0.5252	99.381	99.778	99.683

TABLE 6. Regression Adjusted Estimates DTU Utilization

λ_2	$E[W^*]$	β_0	$\bar{W}^*$	$\bar{W}$	$\widehat{E[W]}_r$
2.00	99.000	−1.0431	98.735	91.282	95.498
1.00	87.000	−0.9934	85.807	79.019	80.204
0.50	49.600	−0.8603	49.283	45.805	46.077

TABLE 7. Assessment of Control and Regression

λ_2	CPU Utilization		DTU Utilization	
	M	V	M	V
2.00	63.464	0.00294	95.500	0.00987
1.00	93.396	0.08334	80.570	0.09072
0.50	99.717	0.00127	46.118	0.00357

$$E[W]_r^{(i)} = \bar{W}_i + \hat{\beta}_{0i}(\bar{W}_i^* - E[W^*]).$$

ith regression adjusted estimate ($i = 1, 2, \ldots, m = 20$)

$$M=\frac{\sum_{i=1}^{20} E[W]_r^{(i)}}{20}$$

$$V=\frac{\sum_{i=1}^{20} (E[W]_r^{(i)}-M)^2}{19}.$$

(11)
$$\hat{\beta}_0=-\frac{\sum_{j=1}^{n} (W_j-\bar{W})(W_j^*-E[W^*])}{\sum_{j=1}^{n} (W_j^*-\bar{W}^*)^2}.$$

since in our case $\mathrm{Var}[W^*]$ is not known. As in the case of straight control, we also display an estimate V of $\mathrm{Var}\{\widehat{E[W]}_r\}$ obtained from a set of $m=20$ independent observations of $\widehat{E[W]}_r$ along with M, the mean of the m observations. Since the bias of the control and regression method appears to be small, comparison of Tables 4 and 7 suggests that an improvement over straight control is obtainable.

A third method of estimation studied in this paper incorporates the notion of antithetics with regression adjusted control. For $i=1, 2, \ldots, 25$, antithetic estimates $W_i^{(1)}$ and $W_i^{(2)}$ are averaged to obtain W_i, an estimate of $E[W]$, and antithetic estimates $W_i^{*(1)}$ and $W_i^{*(2)}$ are averaged to obtain an estimate of $E[W^*]$. Then regression adjusted control of the form (11) is applied to the W_i and W_i^*, the antithetic and regression adjusted estimate of $E[W]$ being

(12)
$$\widehat{E[W]}_{r,a}=\bar{W}+\hat{\beta}_0(\bar{W}^*-E[W^*]).$$

where

$$\bar{W}=\frac{\sum_{i=1}^{n} W_i}{n} \text{ and } \bar{W}^*=\frac{\sum_{i=1}^{n} W_i^*}{n}.$$

Some results of experiments on this technique are reported in Tables 8, 9, and 10. Note, however, that each estimate, $\widehat{E[W]}_{r,a}$, results from the computational work of 100 realizations (50 for the overhead model), whereas each estimate $E[W]_r$ for straight control and $E[W]_r$ for regression adjusted control results from the computational work of but 50 realizations. The gain, if any, obtained from the antithetic device along with regression adjusted control thus appears to be small.

TABLE 8. Antithetic and Regression Adjusted Estimates CPU Utilization

λ_2	$E[W^*]$	$\hat{\beta}_0$	$\bar{W}^*$	$\bar{W}$	$\widehat{E[W]}_{r,a}$
2.00	49.900	−0.9217	51.491	64.898	63.432
1.00	87.000	−0.7996	87.581	93.463	92.999
0.50	99.200	−0.4796	99.257	99.759	99.732

TABLE 9. Antithetic and Regression Adjusted Estimates DTU Utilization

λ_2	$E[W^*]$	$\hat{\beta}_w$	$\bar{W}^*$	$\bar{W}$	$\widehat{E[W]}_{I,R}$
2.00	99.900	−1.0726	98.625	91.185	95.553
1.00	97.000	−0.9321	85.832	79.135	89.165
0.50	49.600	−0.8526	49.780	46.296	46.142

TABLE 10. Assessment of Antithetics With Control and Regression

λ_2	CPU Utilization		DTU Utilization	
	M	V	M	V
2.00	63.447	0.00336	95.546	0.01600
1.00	95.232	0.01469	80.486	0.03228
0.50	99.716	0.00109	46.124	0.00128

$$E[W]_{I,R}^{(i)} = \bar{W}_i + \hat{\beta}_w(\bar{W}_i^* - E[W^*]),$$

ith antithetic and regression adjusted estimate ($i = 1, 2, \ldots, n = 20$)

$$M = \frac{\sum_{i=1}^{n} E[W]_{I,R}^{(i)}}{20}$$

$$V = \frac{\sum_{i=1}^{n} (E[W]_{I,R}^{(i)} - M)^2}{19}.$$

It has been noted that the regression adjusted estimate using (10) or (11) is biased. In order to remove this bias we also computed unbiased regression estimates as follows:

(13) $$E[W]_i = \frac{1}{n}\sum_{i=1}^{n} (W_i + \beta_i(W_i^* - E[W^*])),$$

where

$$\beta_i = -\frac{\sum_{k \neq i} (W_k - \bar{W}_i)(W_k^* - E[W^*])}{\sum_{k \neq i} (W_k^* - \bar{W}_i^*)^2},$$

$$\bar{W}_i = \frac{\sum_{k \neq i} W_i}{n-1}$$

and

$$W_i^* = \frac{\sum_{i \neq j} W_i^*}{n-1}$$

The above unbiased estimates of CPU utilization (Table 11) and DTU utilization (Table 12) were consistently lower than the estimates reported for control and regression reported by Tables 5 and 6, but the actual differences observed were small.

TABLE 11. Unbiased Regression Estimates, CPU Utilization

λ_2	$E[W^*]$	$\bar{\beta}$	$\bar{W}^*$	$\bar{W}$	$E[\bar{W}']$
2.00	49.900	−0.9291	52.130	65.523	63.450
1.00	87.000	−0.6470	88.379	91.090	93.192
0.50	99.200	−0.5229	99.381	99.778	99.669

TABLE 12. Unbiased Regression Estimates, DTU Utilization

λ_2	$E[W^*]$	$\bar{\beta}$	$\bar{W}^*$	$\bar{W}$	$E[\bar{W}']$
2.00	99.900	−1.4031	98.735	94.282	95.498
1.00	87.00	−0.9933	85.807	79.019	80.227
0.50	49.600	−0.8604	49.283	45.805	46.072

REFERENCES

[1] Gaver, D. P., "Statistical Methods for Improving Simulation Efficiency," Proceedings of Third Annual Conference on Applications of Simulation, Los Angeles (Dec. 1969).

[2] Kuehner, C. J. and B. Randell, "Demand Paging in Perspective," Proc. FJCC (1968), pp. 1011–1018.

[3] Lewis, P. A. W. and G. S. Shedler, "A Cyclic-Queue Model of System Overhead in Multiprogrammed Computer Systems," IBM Research Report RC-2813 (Mar. 1970), to appear J. ACM (Apr. 1971).

[4] Page, E. S., "On Monte Carlo Methods in Congestion Problems: II," Operations Research, 13, 300–305 (Mar.–Apr. 1965).

[5] Randell, B. and C. J. Kuehner, "Dynamic Storage Allocation Systems," C.A.C.M., 11, 297–305 (1968).

[6] Shedler, G. S., "A Cyclic-Queue Model of a Paging Machine," IBM Research Rept. RC-2814 (Mar. 1970).

[7] Smith, J. L., "Multiprogramming Under a Page on Demand Strategy," C.A.C.M., 10, 636–646 (1967).

MODELS FOR MULTI-ITEM CONTINUOUS REVIEW INVENTORY POLICIES SUBJECT TO CONSTRAINTS*

D. A. Schrady and U. C. Choe†

Naval Postgraduate School
Monterey, California

ABSTRACT

Models are formulated for determining continuous review (Q, r) policies for a multi-item inventory subject to constraints. The objective function is the minimization of total time-weighted shortages. The constraints apply to inventory investment and reorder workload. The formulations are thus independent of the normal ordering, holding, and shortage costs. Two models are presented, each representing a convex programming problem. Lagrangian techniques are employed with the first, simplified model in which only the reorder points are optimized. In the second model both the reorder points and the reorder quantities are optimized utilizing penalty function methods. An example problem is solved for each model. The final section deals with the implementation of these models in very large inventory systems.

1. INTRODUCTION

Inventories exist to provide service to customers by satisfying their demands from on-hand material.‡ It follows that a reasonable objective of inventory management is the maximization of service provided, which is achieved by minimizing stockouts. In particular, the minimization of total time-weighted shortages is thought to be a desired objective.

In pursuing this objective, the manager of a realistically large, multi-item inventory system has a number of constraints imposed on his "when to buy and how much to buy" decisions. The stock points of the Naval supply system have investment and reorder workload constraints which are real and binding.

The classic variable cost minimization formulation is the most used method for determining inventory policies. Multi-item problems are usually solved by assuming that they can be dealt with as a series of independent, single item problems. In the presence of binding constraints on a population of items this approach is not applicable. Additionally, the cost minimizaton formulation requires the use of cost parameters which are arbitrary or at least very difficult to estimate.

As a consequence of this argument, a series of models are formulated for multi-item, continuous review inventory policies subject to investment and reorder workload constraints. These models do not employ the standard ordering, holding, and shortage costs. This approach was suggested by Tully [8].

In the next section the problem formulation is developed. Section 3 presents a simplified multi-item formulation in which only the reorder points are decision variables. The general multi-item model is developed in section 4. Each model section contains an analysis of the formulation, a solution algorithm, and an example problem. The final section addresses the implementation of the two multi-item models in very large inventory systems.

*This work was supported by the Naval Supply Systems Command under NAVSUP RDT&E work request WR-0-5037.
†Presently, Naval Headquarters, Seoul, Korea.
‡We are ignoring economic motives such as stockpiling in anticipation of price increases.

2. FORMULATION

It is desired to formulate inventory decision rules for multi-item inventories subject to specific constraints. The rules will be of the reorder point, reorder quantity, continuous review type. We assume that all demand which occurs when the on-hand stock is zero is backordered. As suggested in the introduction, the formulation to be used involves the minimization of total time-weighted shortages subject to: (i) total average investment cost less than or equal to an investment limit, and (ii) total number of orders placed per unit time less than or equal to a reorder workload limit.

The specific form of the model depends upon the assumptions about the item demand characteristics and the expressions used for the total average on-hand inventory level, total number of buys per unit time, and total time-weighted shortages per unit time. The first assumption is that the distribution of lead time demand is normal (μ_i, σ_i^2) for all items. The following notation is used throughout the paper. For the i^{th} item let:

c_i = item unit cost in dollars,
λ_i = mean demand per unit time in units;
μ_i = mean leadtime demand in units;
σ_i = standard deviation of leadtime demand in units;
$\Phi_i(r_i)$ = probability that leadtime demand exceeds r_i;
r_i = reorder point; and
Q_i = reorder quantity.

Also let

K_1 = investment limit in dollars, and
K_2 = reorder workload constraint.

With a continuous review inventory policy an order is placed after the demand of Q units of stock. It follows then that the expected number of orders placed per unit time is λ/Q. For a multi-item inventory with N items, the total expected number of orders placed per unit time is

$$\sum_{i=1}^{N} \frac{\lambda_i}{Q_i}.$$

Inventory investment is the priced-out value of the total expected on-hand inventory. As shown by Hadley and Whitin [5] with continuous review the expected on-hand quantity, $E(OH)$, is given by

$$E(OH) = r + \frac{Q}{2} - \mu + B(Q, r),$$

where $B(Q, r)$ is the expression for the expected shortages at any point in time. If lead time demand is normally distributed it can be shown [5] that

$$B(Q, r) = \frac{1}{Q}\left[\beta(r) - \beta(r+Q)\right], \tag{1}$$

where

$$\beta(r) = \frac{1}{2}\left[\sigma^2 + (r-\mu)^2\right]\Phi\left(\frac{r-\mu}{\sigma}\right) - \frac{\sigma}{2}(r-\mu)\phi\left(\frac{r-\mu}{\sigma}\right), \tag{2}$$

$$\phi(x)=\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}, \text{ and}$$

$$\Phi(r)=\int_r^{\infty}\phi(x)dx.$$

The expected on-hand quantity expression can be simplified by omitting the $B(Q, r)$ term and this approximation is reasonable if the risk of stockout is not too large. This assumption is employed throughout the paper. With this assumption the total inventory investment is then given by the expression

$$\sum_{i=1}^{N} c_i\left(r_i+\frac{Q_i}{2}-\mu_i\right).$$

The expected number of backorders at any point in time may be explicitly determined from the steady state probability distribution for negative net inventory levels. Hadley and Whitin [5] used this approach and showed that when leadtime demand has a normal distribution, the time-weighted shortages expression is given by Equation (1). If the risk of stockout is not large, then the expression for time-weighted shortages can be simplified by ignoring the $\beta(r+Q)$ term, yielding the expected time-weighted units short per unit time for the ith item as

$$Z_i(Q_i, r_i)=\frac{\beta_i(r_i)}{Q_i}.$$

Let $Q=(Q_1, Q_2, \ldots, Q_N)$ and $r=(r_1, r_2, \ldots, r_N)$. The multi-item problem formulation can now be given as:

(3) $$\min Z(Q, r)=\sum_{i=1}^{N} Z_i(Q_i, r_i)=\sum_{i=1}^{N}\frac{\beta_i(r_i)}{Q_i},$$

subject to

(4) $$g_1(Q, r)=K_1-\sum_{i=1}^{N} c_i\left(r_i+\frac{Q_i}{2}-\mu_i\right)\geq 0,$$

(5) $$g_2(Q, r)=K_2-\sum_{i=1}^{N}\frac{\lambda_i}{Q_i}\geq 0, \text{ and}$$

$$Q_i\geq 0$$

r_i unrestricted, $i=1, 2, \ldots, N$.

3. SIMPLIFIED MULTI-ITEM MODEL

The basic, multi-item, continuous review formulation was given in the previous section. Suppose here that the order quantities are determined from some other criterion. Specifically, the assumption is made that order quantities are determined from the equation

(6) $$Q_i=h\sqrt{\frac{\lambda_i}{c_i}},$$

which is a form of the Wilson economic lot size formula. The quantity h is a constant which is assumed to be the same for all items.

From the reorder constraint and Equation (6), we then write

$$\sum_{i=1}^{s} \frac{\sqrt{\lambda_i c_i}}{h} = K_2, \text{ or}$$

$$h = \frac{\sum_{i=1}^{s} \sqrt{\lambda_i c_i}}{K_2}. \tag{7}$$

The determination of h then fixes the order quantities from Equation (6) and eliminates one set of decision variables from the problem, i.e.,

$$Q_i = \frac{\sum_{i=1}^{s} \sqrt{\lambda_i c_i}}{K_2} \cdot \sqrt{\frac{\lambda_i}{c_i}}.$$

When these order quantities are used, the investment constraint becomes

$$g_1(r) = K_1 - \sum_{i=1}^{s} c_i \left(r_i + \frac{h}{2}\sqrt{\frac{\lambda_i}{c_i}} - \mu_i\right) \geq 0,$$

where $r = (r_1, r_2, \ldots, r_s)$.

The investment constraint above can be rewritten to the simplified form

$$g_1(r) = K_1' - \sum_{i=1}^{s} c_i r_i \geq 0,$$

where

$$K_1' = K_1 - \frac{h}{2}\sum_{i=1}^{s} \sqrt{\lambda_i c_i} + \sum_{i=1}^{s} c_i \mu_i.$$

The multi-item problem with fixed order quantities can now be stated as:

$$\text{minimize } Z(r) = \sum_{1}^{s} Z_i = \sum_{i=1}^{s} \frac{\beta_i(r_i)}{h}\sqrt{\frac{c_i}{\lambda_i}}, \tag{8}$$

subject to

$$g_1(r) = K_1' - \sum_{i=1}^{s} c_i r_i \geq 0, \tag{9}$$

where h is determined from Equation (7) and r_i unrestricted.

In deriving a solution for the problem as stated in Equations (8) and (9), we set up the Lagrangian function

$$L(r, \eta) = \sum_{i=1}^{s} \frac{\beta_i(r_i)}{h}\sqrt{\frac{c_i}{\lambda_i}} + \eta\left[\sum_{i=1}^{s} c_i r_i - K_1'\right].$$

Taking partial derivatives with respect to the decision variables and setting these expressions equal to zero yields:

$$\frac{\partial L}{\partial r_i}=\frac{1}{h}\sqrt{\frac{c_i}{\lambda_i}}\left[(r_i-\mu_i)\Phi\left(\frac{r_i-\mu_i}{\sigma_i}\right)-\sigma_i\phi\left(\frac{r_i-\mu_i}{\sigma_i}\right)\right]+\eta c_i=0$$

and

$$\frac{\partial L}{\partial \eta}=\sum_i^n c_i r_i-K_1'=0.$$

Thus the necessary conditions for optimality are:

(10) $$\eta=\alpha_i(r_i)(h\sqrt{c_i\lambda_i})^{-1}$$

and

(11) $$\sum_i^n c_i r_i=K_1'.$$

where

$$\alpha(r)=-\frac{\partial}{\partial r}[\beta(r)]=\sigma\phi\left(\frac{r-\mu}{\sigma}\right)-(r-\mu)\Phi\left(\frac{r-\mu}{\sigma}\right).$$

From Kuhn-Tucker theory [4], if we have a convex objective function and a convex constraint region, the necessary conditions are also sufficient. Since the constraint under consideration is linear in the r_i's, the region is convex. To show that the objective function is convex, consider the equation of the expected time-weighted shortages.

$$Z_i=\frac{1}{h}\sqrt{\frac{c_i}{\lambda_i}}\beta_i(r_i).$$

If $\frac{\partial^2 Z_i}{\partial r_i^2}\geq 0$ for all values of r_i, then Z_i is convex. Taking derivatives we obtain

$$\frac{\partial Z_i}{\partial r_i}=-\frac{1}{h}\sqrt{\frac{c_i}{\lambda_i}}\alpha_i(r_i)<0 \quad \text{for all } r_i \text{ values, and}$$

$$\frac{\partial^2 Z_i}{\partial r_i^2}=\frac{1}{h}\sqrt{\frac{c_i}{\lambda_i}}\Phi\left(\frac{r_i-\mu_i}{\sigma_i}\right)\geq 0 \quad \text{for all } r_i \text{ values.}$$

As the second partial is nonnegative, Z_i is convex. It follows that the objective function Z is convex since it is the sum of convex functions.

In general, Equations (10) and (11) cannot be solved in closed form. A numerical solution procedure is suggested. Note that the right side of Equation (10),

$$\eta=\alpha_i(r_i)(h\sqrt{c_i\lambda_i})^{-1},$$

has a lower bound of zero: i.e., $\eta\geq 0$ since $h>0$, $c_i\lambda_i>0$, and $\alpha_i(r_i)\geq 0$ for all values of r_i. Now $\eta=0$ when r_i becomes infinite since $\alpha_i(\infty)=0$, but $r_i=\infty$ violates the investment constraint.

For an initial value of η, it is reasonable to start with $r_i=0$ yielding

$$\eta=\alpha_i(0)(h\sqrt{c_i\lambda_i})^{-1}, \text{ for all } i, \text{ or}$$

$$\eta\leq h^{-1}\min_i[(c_i\lambda_i)^{-\frac{1}{2}}\alpha_i(0)].$$

Let

$$\delta = k^{-1} \min_i [(c_i\lambda_i)^{-1/2}\alpha_i(0)]. \tag{12}$$

Then $\eta = \delta$ implies that there is at least one r_i at zero.

A convenient starting point for our numerical solution is $\eta = \delta/2$. Starting at $\eta = \delta/2$, solve Equation (10) for each of the r_i's. (Note that solution of Equation (10) for a fixed η requires numerical methods.) Then compute the value of the constraint by utilizing the r_i's just determined. Let

$$\sum_{i=1}^{n} c_i r_i = H.$$

Using a bisection search, if $H > K_1'$, increase η by $\delta/4$. If $H < K_1'$, decrease η by $\delta/4$. Recompute the r_i's and the value of

$$\sum_{i=1}^{n} c_i r_i.$$

If the increase (or decrease) of η has not caused the inequality, $H > K_1'$ or $H < K_1'$, to change, increase (or decrease) η by the same amount, $\delta/4$. If the sign of the inequality has changed then reduce the increment to $\delta/8$ and decrease (or increase) η, and solve for the r_i's at each value of η and computing the value of H until the sign of the inequality switches. Successive increments of η are $\delta/16$, $\delta/32$, etc. Continue until $H = K_1'$ or until H is within some tolerable limit of K_1'. This search converges rapidly. The same type of search is used to solve Equation (10) for a fixed value of η, using, in this case, increments of σ, $\sigma/2$, $\sigma/4$, etc., from an initial value of $r = \mu$.

EXAMPLE. Let the multi-item inventory consist of three items. It is assumed that the distribution of lead time demand is normal with mean μ_i and variance σ_i^2 for the ith item. The item date is as follows:

	Item 1	Item 2	Item 3
λ_i	1,000	1,500	2,000
c_i	1	10	20
μ_i	100	200	300
σ_i	100	100	200

Also let $K_1 = \$8{,}000$ and $K_2 = 15$.

From Equation (7) it can be determined that $k = 23.61$ and, from Equation (6) that the order quantities are $Q_1 = 747$, $Q_2 = 289$, and $Q_3 = 236$. The problem formulation can now be stated as

$$\text{minimize } Z(r) = \sum_{i=1}^{3} \frac{1}{23.61} \sqrt{\frac{c_i}{\lambda_i}}\, \beta_i(r_i).$$

subject to

$$g_1(r) = 11{,}921 - \sum_{i=1}^{3} c_i r_i \geq 0.$$

Using δ as defined by Equation (12), the initial value of the multiplier is $\eta = \delta/2 = 0.0318$. Utilization of the dual bisection search produces rapid convergence to the following results:

$$\eta^* = 0.0055,$$
$$r_1^* = 234,$$
$$r_2^* = 264, \text{ and}$$
$$r_3^* = 453.$$

At these values

$$\sum_{i=1}^{3} c_i r_i = 11{,}938,$$

which is within 0.2 percent of K_1. Total time-weighted shortages are

$$Z = \sum_{i=1}^{3} Z_i = 0.218 + 2.763 + 10.523 = 13.501 \text{ unit years of shortage per year.}$$

4. GENERAL MULTI-ITEM MODEL

The problem originally formulated was

$$\text{minimize } Z(Q, r) = \sum_{i=1}^{n} Z_i(Q_i, r_i) = \sum_{i=1}^{n} \frac{\beta_i(r_i)}{Q_i}, \tag{3}$$

subject to

$$g_1(Q, r) = K_1 - \sum_{i=1}^{n} c_i\left(r_i + \frac{Q_i}{2} - \mu_i\right) \geq 0, \tag{4}$$

$$g_2(Q, r) = K_2 - \sum_{i=1}^{n} \frac{\lambda_i}{Q_i} \geq 0, \tag{5}$$

$Q_i \geq 0$, r_i unrestricted, where $\beta_i(r_i)$ was defined in section 2, Equation (2), and $Q = (Q_1, Q_2, \ldots, Q_n)$ and $r = (r_1, r_2, \ldots, r_n)$.

In approaching the solution of this problem, Lagrangian techniques could again be employed, but this leads to a difficult problem involving the decision variables and two multipliers. Because of these difficulties we were led to the penalty function approach for constrained nonlinear optimization.

The penalty function method is based on the minimization of a new function,

$$P(Q, r, \rho) = \sum_{i=1}^{n} \frac{\beta_i(r_i)}{Q_i} - \rho \sum_{j=1}^{2} \ln g_j(Q, r),$$

over a strictly monotonic decreasing sequence of ρ-values $\{\rho_m\}$. Under certain conditions there exists a sequence of feasible points $\{Q(\rho_m), r(\rho_m)\}$ that respectively minimize $\{P(Q, r, \rho_m)\}$, and have the property that $(Q(\rho_m), r(\rho_m)) \to (Q^*, r^*)$—a solution of the original problem—as $\rho_m \to 0$ $(m \to \infty)$. Thus the original constrained minimization is transformed into a sequence of unconstrained minimizations which converge to the minimum of the original problem.

The conditions under which the sequence of unconstrained minimizations converges to the solution of the original problem are given by Fiacco and McCormick ([2] p. 602). Applied to our problem the basic theorem states that if (1) the feasible solution space is nonempty, (2) the objective function is convex and the constraints are concave and both are twice continuously differentiable, and (3) if the penalty function is strictly convex for all $m > 0$, then (i) the penalty function has a unique minimum for every $m > 0$, and (ii) in the limit as $m \to \infty$ and $\rho_m \to 0$, the unconstrained minimum is equal to the minimum of the constrained problem.

The first condition requires that the investment and reorder constraints together represent a feasible problem. The third condition is satisfied if the second condition is satisfied. Thus everything hinges on the convexity of the formulation.

Equation (4) is linear in Q and r and Equation (5) is concave in Q. Together these constraints form a convex region of feasible solutions. The objective function, Equation (3), is convex if its Hessian is positive semidefinite. The Hessian of $Z_i(Q_i, r_i)$ is

$$\nabla^2 Z_i = \begin{bmatrix} \dfrac{\Phi_i(r_i)}{Q_i} & \dfrac{\alpha_i(r_i)}{Q_i^2} \\ \dfrac{\alpha_i(r_i)}{Q_i^2} & \dfrac{2\beta_i(r_i)}{Q_i^3} \end{bmatrix}.$$

where

$$\alpha(r) = -\frac{\partial}{\partial r}[\beta(r)] = \sigma\phi\left(\frac{r-\mu}{\sigma}\right) - (r-\mu)\Phi\left(\frac{r-\mu}{\sigma}\right).$$

The elements of the Hessian are nonnegative for $Q \geq 0$ and all r. The determinant $|\nabla^2 Z_i|$ is evaluated as

$$|\nabla^2 Z_i| = \frac{1}{Q_i^4} f(r_i).$$

where $f(r) = 2\beta(r)\Phi(r) - \alpha^2(r)$. The determinant is nonnegative if $f(r)$ is nonnegative.

Proceeding in the manner suggested by Brooks and Lu [1], the derivative of $f(r_i)$ is determined to be

$$\frac{df(r_i)}{dr_i} = -\beta_i(r_i)\phi_i(r_i).$$

which is negative for all values of r_i since $\beta_i(r_i)$ and $\phi_i(r_i)$ are positive for all finite values of r_i.* It follows that $f(r_i)$ is nonincreasing for all r_i. Further it can be seen that $\lim_{r_i \to \infty} f(r_i) = 0$, so this together with the fact that $f(r_i)$ is nonincreasing implies that $f(r_i) \geq 0$ for all r_i. Thus the determinant of $\nabla^2 Z_i$ is positive semidefinite and the function $Z_i(Q_i, r_i)$ is convex. It follows that the objective function $Z(Q, r)$, which is the sum of convex functions, is convex.

The computational algorithm proceeds as follows. Begin with an initial feasible solution, $\{Q_0, r_0\}$. Select an initial ρ-value, dependent upon $\{Q_0, r_0\}$. Minimize the unconstrained P-function. Iterate on ρ-values using $\rho_{m+1} = \rho_m/d$, where $d > 1$. Terminate computations if the bounds created by the primal and dual solution values satisfy a preselected convergence criterion.

---

*Clearly $\phi_i(r_i)$, the normal density function, is nonnegative. The time-weighted shortages term is by definition nonnegative. To show this, apply the derivative-limit argument twice more, first on the $\beta_i(r_i)$ term and then on its derivative $-\alpha_i(r_i)$, yielding finally a $\Phi_i(r_i)$ term whose sign is clearly nonnegative.

The basic mechanics of the algorithm are given in Reference [2], extensions and the use of extrapolation for accelerating convergence are given in Reference [3], and computational experience is presented in chap. 8 of Reference [4]. A computer code called SUMT, Reference [7], (Sequential Unconstrained Minimization Technique) is available for IBM 360 and CDC 6600 machines.

Example. We employ the same three-item problem that was given in section 3. The difference of course, is that the problem was previously treated in a simplified manner using fixed order quantities while both the order quantities and reorder points are decision variables in the present treatment of the problem.

The problem was run on the Research Analysis Corporation CDC 6600 computer. Starting with the feasible value $Q_1=600$, $r_1=200$, $Q_2=270$, $r_2=260$, $Q_3=300$, and $r_3=400$, the initial solution has the value $Z=17.808$ unit years of shortage per year. The sequence of iterations proceeded as follows:

Iteration	ρ	Z	Q_1	r_1	Q_2	r_2	Q_3	r_3
1	1.0000	14.8199	589.91	244.02	268.75	265.99	305.08	416.53
2	0.0625	13.1337	537.59	252.11	247.39	276.18	286.57	435.13
3	0.0039	13.0175	533.44	252.73	245.70	276.96	285.13	436.52
4	0.000244	13.0102	533.18	252.77	245.60	277.01	285.04	436.61
5	0.000015	13.0097	533.16	252.78	245.60	277.01	285.04	436.61

The iterations converged rapidly and the computer time was small (1.6 sec).

5. IMPLEMENTATION

Two models have been presented for multi-item inventory control under continuous review. Both models were formulated in terms of operational constraints thought to be realistic of actual operations (at least in military supply systems). A realistic formulation is a necessary first step, but the models must be capable of implementation by the inventory system which they seek to represent. A multi-item inventory in that system could comprise anywhere from 3,000 to 70,000 items. Directly employing either of the models presented with an inventory of 70,000 items is not anticipated. It is suggested that the inventory analyst work with a sample of the items and the appropriately-scaled values of the constraints. After obtaining a solution of the "sample" problem, the results would be interpreted in some way so as to produce stocking policies for all the items in the inventory system. Schemes for determining policies for every item in the inventory based on the solution of a sample problem will be developed for each model.

The first model, presented in section 3, represented a simplified treatment of the general formulation in that the order quantities were not optimized. The model, because of its simplicity, is easily implemented. First select a representative sample from the population of items. Item demand and unit cost are probably the most important characteristics to consider in deciding whether or not a given sample is representative. The simplified model of section 3 would then be solved using the sample inventory and constraint values scaled down in proportion to the sample size. The solution of this problem yields order quantities and reorder points for the sample items, but, more importantly, yields values

of the two constants needed to generate policies for all of the individual population items. The quantity h, which is used to determine order quantities from Equation (6), can be interpreted as the ratio of the square root of twice the ordering cost to the inventory holding cost, as imputed by the reorder workload constraint.

The other constant determined by solution by the sample problem is the Lagrange multiplier, η. Reorder points for all of the items in the population are then determined from

$$\alpha_i(r_i) = \eta h \sqrt{\lambda_i c_i},$$

which is a form of Equation (10). The Lagrange multiplier, η, may be interpreted as the shortage cost imputed by the investment constraint.

In summary, the solution of the sample problem determines the constants h and η which are then used to determine (Q, r) policies for all items in the inventory system from equations

$$Q_i = h\sqrt{\frac{\lambda_i}{c_i}} \tag{6}$$

and

$$\alpha_i(r_i) = \eta h \sqrt{\lambda_i c_i} \tag{13}$$

Solution of Equation (13) requires numerical methods of the sort described in section 3. Determination of the appropriate sample size is to some extent dependent on the requirements of the user. In general, sample sizes of between 5 and 10 percent should be satisfactory and should result in feasible computation.

The more general model of section 4 allows for optimization of both the reorder points and reorder quantities. *A priori*, this more general model will yield better policies, but require more computational effort. Successful implementation depends upon efficient computer programs for the sequential unconstrained solution of the sample problem and the subsequent policy calculations for nonsample items. The determination of optimal inventory policies again begins with a sample problem, but with some restriction on the item sample size which will be discussed later. Once a sample has been selected, the SUMT program is utilized to solve the sample problem yielding optimal reorder points and reorder quantities for the sample items. It then remains to use the results of the sample problem to determine policies for each item in the population of items which constitute the inventory system. This step is facilitated by the convergence criterion employed in the RAC SUMT program Reference [7].

Associated with the primal (original) problem there is a dual. The dual employed here is given by Wolfe [9] and is suggested by the Kuhn-Tucker sufficiency conditions for convex programming problems. If we write the primal problem as

$$\min Z(Q, r)$$

$$\text{subject to } g_j(Q, r) \geq 0 \qquad j = 1, 2,$$

then the dual problem is

$$\min L(Q, r, u) = Z(Q, r) - \sum_{j=1}^{2} u_j g_j(Q, r)$$

subject to $\nabla L(Q, r, u)=0$

$$u_j \geq 0.$$

Fiacco and McCormick ([2], p. 602) show that if min $Z(Q, r)=Z^*$, then

$$L(Q(\rho_m), r(\rho_m), u(\rho_m)) \leq Z^* \leq Z(Q(\rho_m), r(\rho_m)).$$

The Lagrange multipliers u_j are related to the SUMT solution of the primal problem by the equation

$$u_j(\rho_m)=-\rho_m \left.\frac{\partial(\ln g_j)}{\partial g_j}\right|_{\rho_m}$$

It is therefore possible, at each iteration in the sequential unconstrained minimization process, to compute the value of the dual. The primal and dual solution values bound the quantity Z^* and may thus be employed in a convergence criterion to terminate the SUMT minimization. The final SUMT solution yields the "optimal" multiplier values. These multipliers can then be used to determine policies for all of the non-sample items in the inventory. To see this we need only write the Lagrangian function for the original problem formulation given in section 2. The Lagrangian would be

$$L(Q, r, u)=\sum_{i=1}^{N}\frac{\beta_i(r_i)}{Q_i}+u_1\left[\sum_{i=1}^{N}c_i\left(r_i+\frac{Q_i}{2}-\mu_i\right)-K_1\right]+u_2\left[\sum_{i=1}^{N}\frac{\lambda_i}{Q_i}-K_2\right].$$

The first order conditions give the optimal reorder quantities and reorder points as

(14)
$$Q_i=\left[\frac{2(u_2\lambda_i+\beta_i(r_i))}{u_1 c_i}\right]^{1/2}$$

and

(15)
$$\alpha_i(r_i)=u_1 c_i Q_i.$$

To summarize then, the SUMT program is used to solve the sample problem and yields optimal policies for the sample item and the optimal multipliers needed to determine optimal policies for the remainder of the population of items.

It should be noted that the solution of Equations (14) and (15) requires iterative procedures. Begin by ignoring the $\beta(r)$ term in Equation (14) giving

$$Q^{(1)}=\left[\frac{2u_2\lambda}{u_1 c}\right]^{1/2}.$$

$Q^{(1)}$ is then used in Equation (15) to determine $r^{(1)}$; this solution is itself by iteration. The resultant value $r^{(1)}$ is then used in Equation (14) to determine $Q^{(2)}$; this solution is straightforward. One continues to iterate until successive Q and r values fail to change significantly. Computational experience has shown convergence to be very rapid.

A proof of convergence is developed as follows. If plotted in (r, Q) coordinates, Equations (14) and (15) would appear as shown in Figure 1. It may be verified that $\frac{dQ}{dr}<0$ and $\frac{d^2Q}{dr^2}>0$ for all values

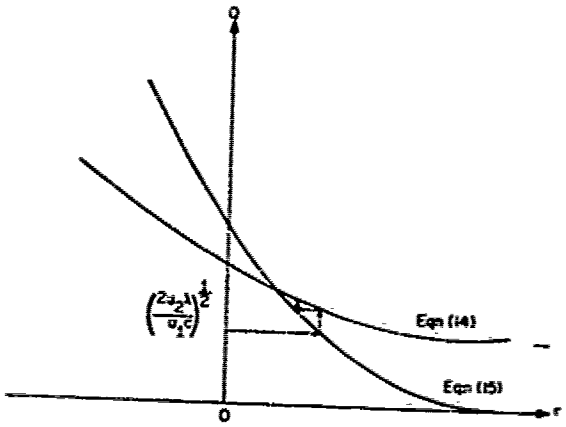


FIGURE 1. Plot of Equations (14) and (15) and indication of the iterative solution

of r, for both Equations (14) and (15). Further, it is clear that Equation (14) is asymptotic to the value $Q=\left(\frac{2u_2\lambda}{u_1c}\right)^{1/2}$ as $r\to\infty$, while Equation (15) is asymptotic to $Q=0$ as $r\to\infty$. If we associate Q_{14} with Equation (14) and Q_{15} with Equation (15), then for large values of r, $Q_{15}-Q_{14}<0$. If we can show that for large negative r values $Q_{15}-Q_{14}>0$, then the curves must cross, indicating the existence of a solution. From Equations (14) and (15) the difference $Q_{15}-Q_{14}$ may be written in the form

$$\alpha(r)-[p+n\beta(r)]^{1/2}, \text{ where } n \text{ and } p$$

are positive constants. It can be shown that in the limit as $r\to-\infty$, $\alpha(r)$ grows as $-r$, while the term $[p+n\beta(r)]^{1/2}$ grows at the rate $\frac{r}{\sqrt{2}}$. Thus for large negative r-values $Q_{15}-Q_{14}>0$. This establishes the existence of at least one simultaneous solution of Equations (14) and (15). Further there is no possibility of converging to a local, non-global solution because the convexity of the original problem, which is preserved in the Lagrangian function, insures that any local solution is in fact a global solution.

Two final comments remain to be made. The first comment involves the size of the sample problem for the model of section 4. The SUMT program, which performs the sequential unconstrained minimization of the P-function, employs the generalized Newton method which requires inversion of the Hessian matrix of the P-function. The Newton move normally requires n^2 storage locations and $n^3/3$ multiplications and additions, where n is the number of decision variables. In our problem $n=2N$, where N is the number of items in the sample problem. Thus the Hessian inversion operation limits the size of the problem which can be reasonably computed with SUMT. However, the structure of our particular formulation can be exploited to reduce the normal matrix inversion storage and computation requirements. McCormick [6] has shown that only $(7N+2)$ storage locations, $14N$ multiplications and additions, and $3N$ divisions are required for the Newton move in the model of section 4. Thus, while still somewhat restricted in size, sample problems of up to $N=500$ items (1,000 variables) should be practical. (The standard SUMT program is restricted presently to 100 variables in consideration of computation time.)

The second comment is concerned with the generation of a first feasible solution for the SUMT computation of the problem of section 4. It seems that solution of the simplified model of section 3

(for the sample problem only) may be the best and quickest way to obtain the necessary first feasible solution. Further, providing a relatively good first feasible solution should minimize the total amount of computation performed by SUMT.

We have presented two models for continuous review control of multi-item inventories. The formulations emphasized operational constraints rather than the classical variable costs postulated to be associated with inventory operations. Either model can successfully be implemented in very large inventories. In closing we note that the investment constraint could be replaced by a limit on total stock replenishment funds, if in some application this constraint was more appropriate, so long as the new constraint is concave in the decision variables.

REFERENCES

[1] Brooks, R. S. and J. Y. Lu, "On the Convexity of the Backorder Function for an E.O.Q. Policy," *Management Science*, Vol. 15, No. 7 (Mar. 1969).

[2] Fiacco, A. V. and G. P. McCormick, "Computational Algorithm for the Sequential Unconstrained Minimization Technique for Nonlinear Programming," *Management Science*, Vol. 10, No. 4 (July 1964).

[3] Fiacco, A. V. and G. P. McCormick, "Extensions of SUMT for Nonlinear Programming: Equality Constraints and Extrapolation," *Management Science*, Vol. 12, No. 11 (July 1966).

[4] Fiacco, A. V. and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* (John Wiley and Sons, Inc., New York, 1968).

[5] Hadley, G. and T. M. Whitin, *Analysis of Inventory Systems* (Prentice-Hall, Englewood Cliffs, New Jersey, 1963).

[6] McCormick, G. P., "SUMT Modifications to Handle Large Structured Inventory Problems," personal communication (12 Feb. 1971).

[7] McCormick, G. P., W. C. Mylander III, and R. L. Holmes, "A User's Manual for Experimental SUMT: The Computer Program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming," Research Analysis Corporation, McLean, Virginia (Mar. 1970).

[8] Tully, A. P., "A Goal-Constraint Formulation for Multi-Item Inventory Systems," Masters Thesis, Naval Postgraduate School, Monterey, California (Dec. 1968).

[9] Wolfe, P., "A Duality Theory for Nonlinear Programming," Quarterly of Applied Mathematics, Vol. XIX, No. 3 (Oct. 1961).

THE BOTTLENECK TRANSPORTATION PROBLEM

R. S. Garfinkel and M. R. Rao

Graduate School of Management
University of Rochester
Rochester, New York

ABSTRACT

The bottleneck transportation problem can be stated as follows: A set of supplies and a set of demands are specified such that the total supply is equal to the total demand. There is a transportation time associated between each supply point and each demand point. It is required to find a feasible distribution of the supplies which minimizes the maximum transportation time associated between a supply point and a demand point such that the distribution between the two points is positive. In addition, one may wish to find from among all optimal solutions to the bottleneck transportation problem, a solution which minimizes the total distribution that requires the maximum time.

Two algorithms are given for solving the above problems. One of them is a primal approach in the sense that improving feasible solutions are obtained at each iteration. The other is a "threshold" algorithm which is found to be far superior computationally.

1. INTRODUCTION

The bottleneck transportation problem as given by Hammer [7] is to minimize

$$z = \max_{(i,j):\, x_{ij}>0} t_{ij}$$

$$\text{(P1)} \qquad \begin{aligned} \sum_{j=1}^{n} x_{ij} &= a_i \qquad i = 1, 2, \ldots, m \\ \sum_{i=1}^{m} x_{ij} &= b_j \qquad j = 1, 2, \ldots, n \\ x_{ij} &\geq 0 \end{aligned}$$

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = W.$$

where a_i = amount available at the i^{th} supply point.
b_j = requirement at the j^{th} demand point.
t_{ij} = transportation time from supply point i to demand point j.
x_{ij} = amount to be transported from the i^{th} supply point to the j^{th} demand point.

and all of the data is integer (or equivalently rational).

P1 belongs to a class known as bottleneck problems [2, 5]. Applications of P1 are given in [7]. A transportation problem can always be written as an assignment problem by appropriately

increasing its size. Hence P1 can be transformed into a bottleneck assignment problem and solved by the algorithm of [5] or [6]: however, such an approach would not be computationally attractive since the size of the problem is increased considerably.

An extension of P1 is also considered in [7]. This problem which we will refer to as P2 can be stated as follows:

Let z^* be the value of an optimal solution to P1. From all optimal solutions to P1, find a solution that minimizes

(P2) $$u = \sum_{(i,j)\,(t_{ij} = z^*)} x_{ij}$$

Hammer [7] provides methods for solving P2 (and thus P1). Corrections to [7] and a thorough review of the East European literature are supplied in [9].

In this paper, we present two methods for solving P2. The first can be viewed as a primal method, in that it generates a sequence of feasible solutions to P1. By solving an appropriate (classical) transportation problem at each iteration, a solution to P2 is found.

In the second method P1 is solved by a "threshold" [2, 5] algorithm. Then, a solution to P2 is found by solving an appropriate transportation problem.

2. PRIMAL ALGORITHM

In this section we present a primal algorithm for solving *P2*. It is similar to the approach of Romanski in [8]. The steps are as given below:

1. Find a starting feasible solution $\hat{X} = (\hat{x}_{ij})$ to *P1*. This can easily be done, for example, by using the well known "North-West Corner Rule."

2. Let $\hat{z} = \max_{(i,j)}\{t_{ij} | \hat{x}_{ij} > 0\}$

and

$$c_{ij} = \begin{cases} 1 \text{ if } t_{ij} = \hat{z} \\ 0 \text{ if } t_{ij} < \hat{z} \\ M \text{ (arbitrarily large) if } t_{ij} > \hat{z}. \end{cases}$$

3. Solve the transportation problem using $C = (c_{ij})$ as the cost matrix. This can be done, for example, by the *u−v method* [1] or the out-of-kilter method [4]. Call this solution $\bar{X}$ and go to Step 4.

4. If the objective function value is zero, go to Step 2. Otherwise, $\bar{X}$, is an optimal solution to *P2*.

It is clear that $\hat{z}$ in Step 2 decreases at each iteration. Thus we need to solve only a finite number (not more than the number of distinct entries in $T = (t_{ij})$) of transportation problems and consequently the procedure is finite.

EXAMPLE

$$T = \begin{array}{|c|c|c|c|c|c|} \hline 7 & 9 & 6 & 4 & 2 & 5 \\ \hline 8 & 13 & 11 & 7 & 3 & 6 \\ \hline 4 & 6 & 9 & 2 & 8 & 8 \\ \hline 9 & 4 & 6 & 9 & 10 & 2 \\ \hline 1 & 3 & 4 & 9 & 8 & 7 \\ \hline \end{array} \quad \begin{matrix} a \\ \\ \\ \\ \\ \\ \end{matrix}$$

$$\begin{array}{ccccc} 7 & 7 & 3 & 4 & 7 \end{array} \quad b$$

$\hat{X}=$

5				
2	4			
	3	3	2	
			2	
				7

$\rightarrow C=$

0	0	0	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Northwest corner solution

$\hat{z}=13=t_{22}$

bottleneck flow = 4

$\hat{X}=$

1	4			
6				
	3	3	2	
			2	
				7

$\rightarrow C=$

0	1	0	0	0
0	M	M	0	0
0	0	1	0	0
1	0	0	1	M
0	0	0	1	0

$\hat{z}=9=t_{12}=t_{22}=t_{44}$

bottleneck flow = 9

$\hat{X}=$

1			4	
6				
	7			1
		2		
		1		6

$\rightarrow C=$

0	M	0	0	0
1	M	M	0	0
0	0	M	0	1
M	0	0	M	M
0	0	0	M	1

$\hat{z}=8=t_{21}=t_{32}=t_{43}$

bottleneck flow = 13

$\hat{X}=$

				5
			4	2
1	7			
		2		
6		1		

$\rightarrow C=$

1	M	0	0	0
M	M	M	1	0
0	0	M	0	M
M	0	0	M	M
0	0	0	M	M

$\hat{z}=7=t_{24}$

bottleneck flow = 4

$\hat{X}=$

			4	1
				6
1	7			
		2		
6		1		

$\rightarrow C=$

M	M	1	0	0
M	M	M	M	0
0	1	M	0	M
M	0	1	M	M
0	0	0	M	M

$\hat{z}=6=t_{32}=t_{33}$
bottleneck flow = 9

$\hat{X}=$

			4	1
				6
7	1			
	2			
	4	3		

$\hat{z}$ remains at $6=t_{33}$
bottleneck flow = 1
optimal solution

3. THRESHOLD ALGORITHM

The algorithm given here generates a sequence of improving lower bonds on z^*. A feasible solution is not obtained until the final iteration.

For each row i of T, let p_i be any permutation of $\{1, 2, \ldots, n\}$ such that

$$t_{i,p_i(1)} \leq t_{i,p_i(2)} \leq \ldots \leq t_{i,p_i(n)}.$$

Let

$$s_{ik}=\sum_{j=1}^{k} b_{p_i(j)}$$

and $s_{ik^*}=\min_k\{s_{ik} \mid s_{ik} \geq a_i\}$. Then $r_i=t_{i,p_i(k^*)}$ is a lower bound for z^*, since the i^{th} supply constraint cannot be satisfied using only cells with time less than r_i.

Similarly, for each column j of T let q_j be a permutation of $\{1, \ldots, m\}$ such that

$$t_{q_j(1),j} \leq t_{q_j(2),j} \leq \ldots \leq t_{q_j(m),j}, s_{kj}=\sum_{i=1}^{k} a_{q_j(i)} \text{ and } s_{k^*j}=\min_k\{s_{kj} \mid s_{kj} \geq b_j\}.$$

Then $r_j=t_{q_j(k^*),j}$ is also a lower bound on z^*.

Finally, compute $z^0=\max\{r_1, \ldots, r_m, c_1, \ldots, c_n\}$ as the best lower bound for z^*.

The algorithm operates by solving a sequence of max-flow problems. The network is the standard one for transportation problems. It has a source s, supply nodes ($i=1, \ldots, m$), demand nodes

$(j=1, \ldots, n)$ and a sink t. Arcs are (s, i) with capacity a_i, (j, t) with capacity b_j, and (i, j) with infinite capacity. Only a subset (see Step 2) of the latter arcs are used, however.

Algorithm

Step 1—Let $F=-\infty$ and $z=z^0$.

Step 2—Let all arcs (ij) with $t_{ij} \leq z$ be admissible. Apply the Ford-Fulkerson [3] labeling algorithm to get a maximum total flow $\hat{F}$ through admissible arcs. If $\hat{F}=W$, then $z^*=z$ is the optimal objective function value for P1. Go to Step 4. If $\hat{F}<W$, go to Step 3.

Step 3—Let $F=\hat{F}$. Let $z=\min\{t_{ij} \mid i \text{ labeled}, j \text{ unlabeled}\}$. Clearly at least one of these arcs must be used in order to get further labeling and allow for the possibility of increased flow. Leave the labels from the previous iteration intact and go to Step 2.

Step 4—Let the arc flows obtained in Step 2 be $\hat{X}=(\hat{x}_{ij})$. Let

$$\hat{u}=\sum_{(ij)\mid t_{ij}=z^*} \hat{x}_{ij}$$

If $\hat{u}=W-F$, then the optimal solution to P2 is $\hat{X}$. If $\hat{u} \neq W-F$ go to Step 5.

Step 5—For all admissible arcs, let

$$c_{ij}=\begin{cases}1 \text{ if } t_{ij}=z^*\\ 0 \text{ if } t_{ij}<z^*.\end{cases}$$

Solve the associated classical transportation problem using only the admissible arcs and cost matrix C. The solution solves P2.

Step 4 of the algorithm needs some explanation. In that step, $\hat{u}$ represents the total flow through the bottleneck arcs when an optimal solution to P1 is obtained in Step 2. W is the total flow required. F is the maximum flow obtained at the previous occurrence of nonbreakthrough.

PROPOSITION: If $\hat{u}=W-F$ in Step 4, then a solution to P2 is given by $\hat{X}$.

PROOF: Assume that there is a solution X' to P2 better than $\hat{X}$. Then $u'<\hat{u}$. Also

$$W-u'>W-\hat{u}=F.$$

This implies that with only arcs $\{(ij) \mid t_{ij}<z^*\}$ as admissible, there is a maximal flow greater than F. This contradicts the fact that a maximal flow is obtained in Step 2.

The transportation problem to be solved in Step 5 includes only arcs (ij) such that $t_{ij} \leq z^*$. A feasible solution to the problem is available from Step 2.

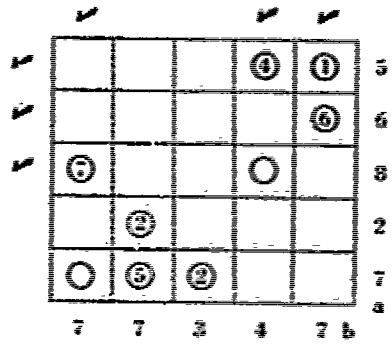
EXAMPLE The 5×5 example of section 2 will be solved.

7	9	6	4	2	5
8	13	11	7	3	6
4	6	9	2	8	8
9	4	6	9	10	2
1	3	4	9	8	7
					2
7	7	3	4	7	1

$z^*=\max\{2, 3, 4, 4, 1, 1, 3, 4, 2, 3\}=4.$

Step 1—$F=-\infty$, $\bar{z}=z^0=4$.

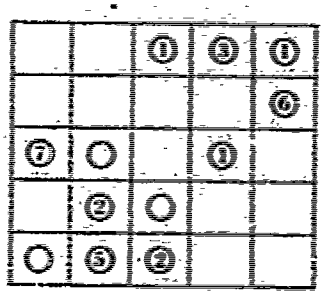
Step 2—max flow iterations yield the following flows with labelled rows and columns checked and admissible arcs circled.



$\hat{F}=27<W=28$.

Step 3—$F=27$, $\bar{z}=6$.

Step 2



$\hat{F}=W$.

Step 4 $\bar{z}=1=W-F$. Terminate, the solution is also optimal to $P2$.

Results

Both methods were programmed in Fortran for the IBM 360/65. Results are shown in Table 1. The computer program for the out-of-kilter method takes advantage of the fact that we always start with a feasible solution and that the lower bound is zero while the upper bound is infinite for all arcs. The primal algorithm using the u-v method is listed as Algorithm 1, with the out-of-kilter method as Algorithm 1a. The threshold method is Algorithm 2. For Algorithms 1 and 1a the northwest corner approach was used for the first feasible solution.

A total of 11 randomly generated problems were run and tested by the three algorithms. The elements of T were uniformly distributed integers between 0 and Q, where Q varied from problem to problem. As Q increases, the upper limit on the number of problems that methods 1 and 1a have to solve also increases. Conversely the individual problems are likely to be less difficult since the number of ones in C will be decreased.

TABLE I

Problem	M	N	E	Q	z^*	u^*	Algorithm 1		Algorithm 1a		Algorithm 2			
							Time (Sec)	Iterations[a]	Time (Sec)	Iterations[a]	Time (Sec)	Iterations[a]	$\bar{u}$	z^*
1	25	25	711	100	23	NC[c]	13	51	55	43	0.4	1	NC	23
2	25	50	1,216	250	39	NC	60	113	205	86	0.69	1	NC	39
3	25	50	1,232	500	101	23	44	138	178	103	1.0	2	23	102
4	25	50	1,379	750	148	6	41	132	191	106	1.2	4	NC	148
5	25	50	1,338	1,000	212	4	37	126	139	102	1.6	2	NC	200
6	50	50	2,668	200	28	4	120	119	[d]	[d]	1.2	2	4	26
7	50	50	2,659	500	51	13	165	207	[d]	[d]	2.7	3	14	51
8	50	50	2,422	1,000	234	12	120	182	[d]	[d]	1.2	1	21	234
9	100	100	5,100	100	8	31	1,228	88	[d]	[d]	4.1	1	322	8
10	100	100	5,031	300	35	NC	>600[d]	[d]	[d]	[d]	7.1	1	886	35
11	100	100	4,972	1,000	92	NC	>1,200[d]	[d]	[d]	[d]	3.1	1	127	92

[a] number of transportation problems solved
[b] number of distinct values of z
[c] not calculated
[d] did not obtain the optimal solution in allotted time

Algorithm 2 was only programmed to solve $P1$, since solving $P2$ corresponds to the last iteration of algorithms 1 and 1a, and an indication of time required to perform such an iteration is available from the results of 1 and 1a. Of course, if $\bar{u} = W - F$ in Step 4, then such an iteration is not required. However, it turned out that $z^* = \bar{z}^0$ was usually true. In this case $\bar{u} = W - F = \infty$. This would indicate that a reasonable approach would be to decrease z^* by one, to ensure *not* getting z^* at the first iteration, but to get a much better estimate of u^*, the optimal u.

Algorithm 1a was clearly less efficient than Algorithm 1, and was only used on Problems 1–5.

5. CONCLUSIONS

The problems considered in [7] belong to a class of problems known as bottleneck problems. It is shown in this paper that the bottleneck transportation problem can be solved by an approach which is primal in the sense that improving feasible solutions are found at each iteration. A "threshold" algorithm is also given for the same problem and is shown to be computationally superior to the other approach.

Note that an alternative primal algorithm is to solve $P1$ first and then $P2$. Instead of solving a sequence of transportation problems one could solve a sequence of flow problems over a decreasing set of admissible arcs and then one transportation problem. This technique would still lack an efficient way of going from one flow iteration to the next (i.e. the labels would have to be erased).

Two problems which are variants of $P1$ and $P2$ are also solved in [7]. The first one is to find among all solutions to $P1$, a solution which minimizes the sum of the transportation costs given by

$$\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij},$$

where d_{ij} is the unit transportation cost from supply point i to demand point j. In the second one it is

required to find among all solutions to $P2$, a solution which minimizes the sum of the transportation costs. These problems are also easily solved by both methods discussed in this paper. The only change that is required is in the objective function of the corresponding transportation problem(s) solved by each method. Needless to say, we would expect the "threshold" algorithm to be better than the primal approach for these problems also.

REFERENCES

[1] Dantzig, G. B., *Linear Programming & Extensions* (Princeton University Press, Princeton, New Jersey, 1963) pp. 309–313.

[2] Edmonds, J. and D. R. Fulkerson, "Bottleneck Extrema," The Rand Corporation, RM-5375-PR (Jan. 1968).

[3] Ford, L. R., and D. R. Fulkerson, *Flows in Networks* (Princeton University Press, Princeton, New Jersey, 1962).

[4] Fulkerson, D. R., "An Out-of-Kilter Method for Minimal Cost Flow Problems," J. Soc. Indust. & Appl. Math., 9, 18–27 (1961).

[5] Garfinkel, R. S., "An Improved Algorithm for the Bottleneck Assignment Problem," Working Paper Series No. 7006, College of Business Administration, University of Rochester, Rochester, New York, January 1970.

[6] Gross, O., "The Bottleneck Assignment Problem," The Rand Corporation, P-1630 (1959).

[7] Hammer, P. L., "Time-Minimizing Transportation Problems," Nav. Res. Log. Quart. 16, 345–357 (1969).

[8] Niesterow, E. P., *Transportation Problems in Linear Programming* (Moscow, 1962), pp. 72–80 (in Russian).

[9] Szwarc, W., "Some Remarks on the Time Transportation Problem," Nav. Res. Log. Quart. (this issue).

SOME REMARKS ON THE TIME TRANSPORTATION PROBLEM*

Włodzimierz Szwarc

The University of Wisconsin-Milwaukee
Milwaukee, Wisconsin

1. INTRODUCTION

One of the last issues of this journal contains a paper of P. L. Hammer [5] on the so-called Time Transportation Problem (TTP). Since Ref. [5] does not reference the previous work on this particular problem,† it may be worthwhile to give a historical sketch of TTP. Here we shall do this in a way that (a) corrects or amends some of the deficiencies in [5] at the same time that we (b) align these developments with other parts of the pertinent literature.

The Time Transportation Problem (according to the notations of [9]) is a problem of finding an $m \times n$ matrix $X = \{x_{ij}\}$ which satisfies conditions

$$\left.\begin{aligned} \sum_{j=1}^{n} x_{ij} &= a_i \qquad i = 1, \ldots, m \\ \sum_{i=1}^{m} x_{ij} &= b_j \qquad j = 1, \ldots, n \end{aligned}\right\} \tag{1}$$

and minimizes

$$t_X = \max_{(i,j)\in\theta_X} t_{ij}, \tag{2}$$

where

$$\theta_X = \{(i, j) \mid x_{ij} > 0\}. \tag{3}$$

Numbers $a_i > 0$, $b_j > 0$,‡ $t_{ij} \geq 0$ are given and $\sum_i a_i = \sum_j b_j$.

The TTP was posed and solved in 1959 by A. S. Barsow [1]. The solution method was based on the simplex method.

E. P. Niestierow [7] solved this problem by an adaptation of Kantorowitch's linear programming dual method. In [7] there is also given a method by I. W. Romanowski based on the reduction of TTP

---

*This paper was prepared when the author was a member of the faculty of the School of Urban and Public Affairs, Carnegie-Mellon University, Pittsburgh, Pennsylvania

†The only two references mentioned in Ref. [5] do not actually, deal with the TTP itself.

‡Some authors assume $a_i \geq 0$, and $b_j \geq 0$ but equality $a_i = 0$ *automatically* implies

$$\sum_j x_{ij} = 0$$

for this particular i, in which case row i may be removed

to a classical transportation problem with a cost matrix changing in a course of the iterative solution procedure. W. Grabowski [3] and [4]† solved the TTP by transforming the problem into a single classical transportation problem.

The author of this note presented in [8] as well as in [9] a method based on the theory of graphs. This method consists in finding a sequence of basic feasible solutions

$$X_1, X_2, \ldots, X_k, \tag{4}$$

where X_k is the optimal solution of TTP.

The same author proposed a slight modification of this method in [10]. According to the modified method the corresponding sequence

$$t_{X_1}, t_{X_2}, \ldots, t_{X_k} \tag{5}$$

satisfies conditions

$$t_{X_s} \leqslant t_{X_{s+1}} \quad \text{for each } s = 1, 2, \ldots, k-1. \tag{6}$$

S. I. Zukhovitsky and L. I. Avdejeva[11] published two versions of a solution procedure where the elements of (4) may be not basic solutions (some of them may have more than $m+n-1$ positive x_{ij}). Paper [9] contains a proof that the method from [8] produces an optimal solution in a finite number of steps provided (6) holds. This is always the case when the problem is nondegenerate.

A. Janicki [6] set up a remarkably efficient computer program for the method given in [9]. He proved in [6] the finiteness of this method in any case (including the case when (6) doesn't hold). This paper using some idea of [6] offers a new and simple proof that cycling in TTP is impossible (when one solves the TTP by the method from [10]). So there is no need of using perturbation technique for the degenerate cases.

In 1969 P. L. Hammer [5] published a method which is equivalent to the method given in [8] and [9] in the sense that they produce the same sequence (4) of basic feasible solutions provided we start with the same initial solution.*

The theory of the method from [5] is based on three theorems.‡ One of the theorems which supposed to be a justification of the finiteness of the method is not true as will be shown in the next section. Another theorem of [5] concerning the equivalency of a local and global optimum is true, but the proof is incorrect. The correct proof of it where we followed the reasoning of the author of [5] is given in the appendix.

This section contains an outline of the method from [8] including the modification from [10]. These are stated all necessary theorems. The proofs of Theorems 1 and 2 can be found in [9] or [10].

---

†The review of this paper appeared in Mathematical Review, Vol. 32, Part II, No. 9047 (1966).

*And provided we apply the following devise (substep 2.1. from [5], page 347). In case when there are several (i, j) with the maximal t_{ij} corresponding to the positive basic variables x_{ij}, try to remove from the basis the greatest x_{ij}.

‡One of the theorems is actually a statement but of great importance.

whereas Theorem 3 is proved in section 3. The method is illustrated by a numerical example.

R. S. Garfinkel and M. R. Rao [2] proposed in this issue a solution method of TTP for the case when a_i and b_j are natural numbers (or rational), tall previously discussed methods work for arbitrary positive a_i and b_j). This method is based on the Ford-Fulkerson labeling procedure of finding the maximal flow in a network.

2. SOME REMARKS ON PAPER [5]

The solution method of [5] is given as follows:

1. Determine a basic feasible solution.

2. Find an adjacent better basic feasible solution. [This step consists of 4 sub-steps, as indicated below in substeps 2.1 to 2.4.]

3. Perform step 2 until no adjacent basic feasible solution will be better than the considered one.

On page 347 of [5] there is the following statement:

"From the fact that every time step 2 is carried out the value of $p \cdot t$ is reduced by at least 1, it follows that the algorithm produces an optimal solution in a finite number of steps."

The following two examples will show that the first part of this last statement is not true.

Consider the following 4 x 4 TTP where

$T = (t_{ij}) =$

10	7	1	25	4
3	9	13	12	16
8	8	15	5	20
14	9	15	8	11
16	5	18	12	

and where a_i and b_j are written on the right and below the matrix T.

The author of [5] allows us to start with any feasible basic solution. (see page 348-bottom in [5])

Step 1. We start with the following basic feasible solution

$X_1 = (x_{ij}^1) =$

4			
	5		11
12		8	
		10	1

where $B_1 = \{(1,1), (2,2), (2,4), (3,1), (3,3), (3,3), (4,4)\}$. According to (8) in [5] set N consists of cells (3,3), (4,3) (i.e., cells with the max $t_{ij} = 15$).

Proceed to step 2.

Substep 2.1. Find the greatest $x_{ij}, (i,j) \in N$ which is $x_{43} = 10$.

Substep 2.2. Determine $S_{4,3} = \{(2,1), (2,3), (4,1)\}$ (i.e., the set of cells, except for (4,3), which are at the intersection of arrows—see next chapter.

Substep 2.3. Find the element of $S_{4,3}$ with the minimal t_{ij}.

This is (2,1) with $t_{21} = 3$.

Substep 2.4. Introduce (2,1) and according to the transportation technique remove (4,3). The new set B_2 becomes $B_2 = B_1 + \{(2,1)\} - \{(4,3)\}$ and the adjacent solution is

$X_2 =$

4			
10	5		1
2		18	
			11

However, X_2 is not better than X_1 since $t_2 = t_1 = 15$ and $t_2 p_2 = 15 \cdot 18 = t_1 p_1 = 15\ (10+8)$.

Remark 1. If by performing substep 2.1 we chose $x_{33} = 8$ instead of $x_{43} = 10$, where both (3,3) and (4,3) belong to N then step 2 will lead us from X_1 to X_2' which has a smaller $p = 10 \div 4 = 14 < 10 + 8 = 18$.

$X_2' =$

		4	
	5		11
16		4	
		10	1

The next example will show that performing step 2 we may get even a "worse" solution.

Consider a 4 x 4 TTP with the following data:

$T =$

10	7	4	25	4
3	9	13	12	16
8	8	15	5	20
14	9	15	28	10
16	5	18	11	

Now start with

$\bar{X}_1 =$

4			
	5		11
12		8	
		10	0

Here N, S_{kl} and (h_s, k_s) are the same as in the previous example. Performing step 2 via substeps 2.1, 2.2, 2.3, and 2.4, we obtain the following result:

$\bar{X}_2 =$

4			
10	5		1
2		18	
			10

$\bar{X}_2$ is worse than $\bar{X}_1$ since $t_2 = 28 > t_1 = 15$ and also $t_2 p_2 = 28 \cdot 10 = 280 > t_1 p_1 = 15 \cdot (10 \div 8) = 270$.

As we have shown step 2 doesn't imply a decrease of the value $p \cdot t$. This means that the cited statement on page 347 of [5] cannot serve as a proof of the finiteness of the solution procedure.

Remark 2. Let us return to the second example. We may prevent the objective function (2) from increasing by removing $x_{14}^1 = 0$ from the basis. The new basic variable x_{21}^1 will be zero and $X_2 = X_1$. This rule makes the only difference between the modified method in [10] and the original method in [9].

Sequence (5) which satisfies (6) may, however, not strictly decrease.

3. OUTLINE OF TTP METHOD*

Before we start with our own solution procedure, let us introduce some definitions.

Let X be a basic feasible solution where B is a set of (i, j) of all basic variable x_{ij}. We will denote such a solution by $X(B) = \{x_{ij}^0\}$. B is called a feasible basis.

Let $(k, l) \in B$. Consider the set of cells $B = (k, l)$. Link every two nearest cells of this set which are on the same row or column by a segment (link).

As known [9] this set will consist of two disjoint sets Ω_1 and Ω_2 (one of them may be empty), such that no element of either set is linked with an element of the other set.

By Ω_1 we mean either an empty set if (k, l) is the only cell of B in column l or that set which contains a cell in column l.

By I_1 we denote the set of rows, and by J_1 the set of columns of a $m \times n$ rectangular table in which the elements of Ω_1 lie. In a similar way we define the set of rows I_2 and the set of columns J_2 which are determined by Ω_2.

Let I be the set of all rows and J the set of all columns of an $m \times n$ table. Further let

$$\bar{I}_1 = I - I_1, \quad \bar{J}_2 = J - J_2.$$

Let Φ be a set of all cells of a $m \times n$ rectangular matrix. We introduce the set Ψ, $\Psi \subset \Phi$

$$\Psi = \bar{I}_1 \times \bar{J}_2 - (k, l).\dagger \tag{7}$$

Let Π be any subset of Φ.

By a Π solution we mean each solution of TTP that satisfies conditions

$$x_{ij} = 0 \text{ for all } (i, j) \in \Pi$$

*See [8], [9], [10].

†Set Ψ is identical with set $S_{k,l}$ in [4].

Let $X(B)$ be a basic solution where x_{kl}^B is positive (then $(k, l) \in B$). Then following theorem holds (for proof see [9], [10]).

THEOREM 2: If $\Psi \subset \Pi$ then there exist no Π solution $X = \{x_{ij}\}$ whose element $X_{kl} = 0$.†

This theorem serves as an optimality criterion for the TTPmethod.

We present the TTPmethod, which is as follows.

1. Find an initial basic solution $X(B_1)$ by any of the known methods (for example by the minimum row method).

2. Find $t_{X(B_1)} = t_{kl}$. Define Π_1 as follows:

$$\Pi_1 = \{(i, j) \mid (i, j) \neq (k, l),\ t_{ij} \geq t_{kl},\ x_{ij}^B = 0\}$$

and consider from now on Π_1 solutions only.

3. Find the corresponding Ψ. There are two cases:

a) $\Psi \subset \Pi_1$. Then $X(B_1)$ is the optimal solution of TTP.

This follows from Theorem 2.

b) $\Psi\Pi_1 \neq \phi$.* Then proceed to 4.

4. Find $\min t_{ij} = t_{pq}$. Apply the known transportation technique to find a new adjacent basis by introducing to B_2 cell (p, q). There are two cases. The set of cells for which x_{ij} may increase

a) contains an element, say (u, v) of Π_1.

b) does not contain an element of Π_1.

In case 4(a) the new set $B_2 = B_1 + \{(p, q)\} - \{(u, v)\}$ and the new basic solution $X(B_2) \equiv X(B_1)$ and $\Pi_2 = \Pi_1$. In case 4(b) apply the usual transportation technique obtaining a new set $B_2 = B_1 + \{(p, q)\} - \{(r, s)\}$, a new solution $X(B_2)$, and a new set Π_2 where

$$\Pi_2 = \Pi_1 + \{(i, j) \mid t_{ij} \geq t_{X(B_2)}\} - \{(k_2, l_2)\},$$

where $t_{k_2 l_2} = t_{X(B_2)}$ (if $(k, l) \in B_2$ then $(k, l) \equiv (k_2, l_2)$).

5. Repeat steps 2–4 for B_2 by restricting to Π_2 solutions (Π_2 is defined in step 4) and continue the iteration procedure until encountering in (4) a solution satisfying condition a from step 3. According to Theorem 2, this solution is optimal.

In the course of the procedure apply the following rule. Cell (k, l)—once a candidate for the removal from basis B_t (see step 2) and which was not removed from B_t will be the only candidate for removal from B_{t+1}.

THEOREM 3: The solution method defined by steps 1–5 produces an optimal solution in a finite number of iterations.

PROOF

Part 1 *Preliminary remarks and notations*

Note that the solution procedure possesses the following properties

1°. It orders the basic solutions of (4) in such a fashion that

$$t_{X(B_t)} \geq t_{X(B_{t+1})} \text{ and } \Pi_t \subset \Pi_{t+1}.$$

---

†One can prove an even stronger theorem: if $\Psi \subset \Pi$ then there exists no Π solution $\{x_{ij}\}$ whose element x_{kl} is $< x_{kl}^B$.

*$\overline{\Pi}_1 = \Phi - \Pi_1$.

2°. Cell (k, l) (from step 2), and (u, v) (from step 4a) once removed from B_t cannot belong to any following basis B_h, $h > t$ since

$$(k, l) \in \Pi_{t+1}, \text{ whereas } (u, v) \in \Pi_t \subset \Pi_{t+1}.$$

Definition 1. By a route $\{(i, j) \to (\bar{i}, \bar{j})\}$ we mean a set of cells of a $m \times n$ rectangular table which can be arranged in sequence of the following form:

$$(8) \qquad \left.\begin{matrix}(i, j), (i_1, j), (i_1, j_1), (i_2, j_1), \ldots, (\bar{i}, j) \\ \text{or} \\ (i, j), (i, j_1), (i_1, j_1), (i_1, j_2), \ldots, (\bar{i}, j)\end{matrix}\right\},$$

and where no more than two cells appear on one line (row, column).

As known to any pair of elements of $(i, j), (\bar{i}, \bar{j}) \in B$ there exists exactly one route $\{(i, j) \to (\bar{i}, \bar{j})\} \subset B$ (i.e., whose all elements belong to B).

Definition 2. Let (i, j) be an arbitrary node. By a distance $d_B[(i, j), (\bar{i}, \bar{j})]$ we mean the number of elements of sequence (8) whose elements except of possibly $(\bar{i}, \bar{j})$ belong to B.

Consider a basic feasible solution $X(B)$. Introduce set B^0.

Definition 3. $B^0 = \{(i, j) \mid (i, j) \in B, x_{ij}^B = 0\}$.

Let $(k, l) \in B$ and $t_{kl} = T_{X(B)}$. Define $\tilde{B}^0$ as follows:

Definition 4. Cell (i, j) belongs to $\tilde{B}^0$ if $(i, j) \in B^0$ and (i, j) is the only element of B^0 in the route $\{(i, j) \to (k, l)\} \subset B$.

Part II *Main Part of the proof*

As known (e.g., [9]) to each feasible basis there corresponds exactly one basic solution. Therefore if sequence (4) consists of basic solutions where no basis appears twice then (4) (and so the number of iterations) is finite since the number of all bases is less than

$$\binom{mn}{m + n - 1}.^*$$

Assumption A Assume to the contrary that (4) contains an w-element segment ($w \geq 3$) which we will for convenience denote by

$$(9) \qquad X(B_1), \ldots, X(B_w).$$

where all bases $B_1, \ldots, B_w$ are different except of $B_1 = B_w$.

Assumption A and properties 1° and 2° immediately imply

3° $\Pi_1 = \Pi_2 = \ldots = \Pi_w$;

4° all basic solutions of (8) are identical;

5° if $t_{k1} = t_{X(21)}$ then $t_{k2} = t_{X(B2)} = \ldots = t_{X(Bw)}$.

Consider an arbitrary cell $(i, j) \in \tilde{B}_t$. It is easy to see that

$$(10) \qquad d_{B_t}[(i, j), (k, l)] = d (= \text{constant}) \text{ for all } t = 1, \ldots, w.$$

---

*Actually the number of feasible bases is less than $m^{n-1}n^{m-1}$.

These are the cases: a) d is even: and b) d is odd.
In the first case (i,j) taccording to the procedure) cannot be removed from B_t. In the second case (i,j) once removed from B_t cannot enter any subsequent basis, which contradicts assumption A.
Thus we have shown that $\hat{B}_1 = \hat{B}_2 = \ldots = \hat{B}_w \subset B_t$ for $t = 1, \ldots, w$.
Therefore instead of (9) we may consider another sequence

$$X^1(B_1), X^1(B_2), \ldots, X^1(B_w).$$

where $X^1(B_t) = \{x_{ij}^{1B_t}\}$ is defined as follows

$$x^{1B_t} = \begin{cases} x_{ij}^{1B_t} \\ x_{ij}^{1B_t} + 1 \end{cases} \text{for} \begin{matrix} (i,j) \in \hat{B}_t \\ (i,j) \in \hat{B}_t' \end{matrix} \tag{11}$$

It is obvious that (10) consists of identical matrices (see 3°, 4°, 5°).
Similarly we define $\hat{B}_t^1$ and matrix $X^2(B_t)$ and obtain a sequence of identical matrices

$$X^2(B_1), \ldots, X^2(B_w);$$

repeating the same procedure several times we reach a sequence

$$X^b(B_1), \ldots, X^b(B_w),$$

which consists of identical matrices with all basic elements positive.
But then $B_1 = B_2 = \ldots = B_w$ which contradicts assumption A that (9) (and (10)) contains a basis different from $B_1 = B_w$.
Therefore no basis appears in (4) twice. QED.

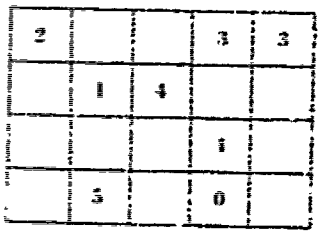
Example.* Consider a 4×5 TTP:

$T = \{t_{ij}\}$

6	21	19	12	7	8
9	13	10	14	15	5
11	11	12	9	12	4
12	16	8	20	19	5
2	6	4	7	3	

The numbers a_i and b_j are on the right and below the matrix T, respectively. Using the minimum row method, we find the initial basic solution $X(B_1)$

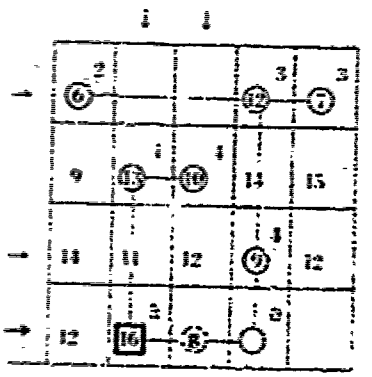
*This example is taken from [10].



Here $t_{x(B_1)} = t_{42} = 16$ and $\Pi_1 = [(1, 2), (1, 3), (4, 4), (4, 5)]$.

We consider from now on only Π_1 solutions.

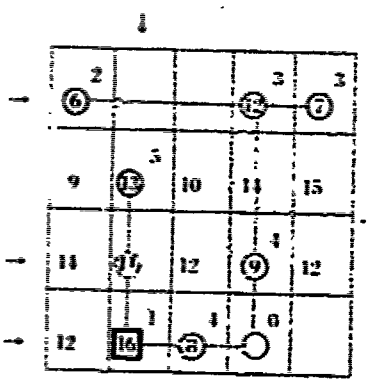
Circle all t_{ij} corresponding to the basic variables. Put the values of basic variables above the circles. Empty cells or circles denote elements of Π_1. Link each two nearest circles which are in the same row and column.



Here $(k, l) = (4, 2)$. Determine Ω_1. This set occupies row 2 and columns 2 and 3.

Consider Ψ (i.e., the set of cells, except (4, 2), which are on the intersection of the row and column arrows).

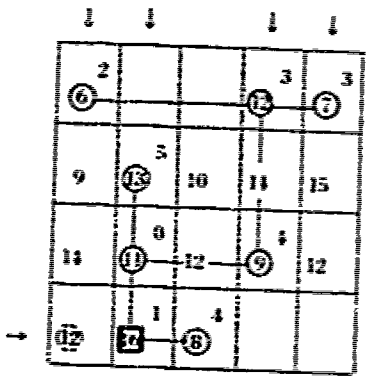
Applying step 4, we find $\min_{(i,j)\in\Psi} t_{ij} = t_{43}$. According to the transportation technique $B_2 = B_1 - \{(2, 3)\} + \{(4, 3)\}$ (since the set of cells for which x_{ij} may increase contains no element of Π_1), and $X(B_2)$ is as follows:



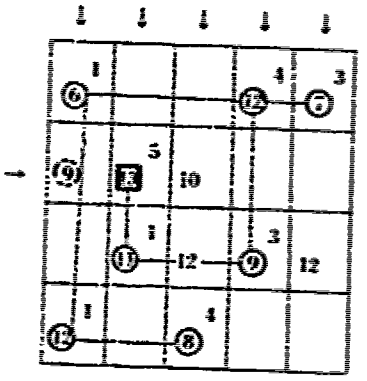
Note that $\Pi_2 = \Pi_1$.

Determine $\Psi = \{(1, 2), (3, 2)\}$ and $\phi\Pi_2 = \{(3, 2)\}$. Here, however, there exists a cell $(4, 4) \in \Pi_2$ for which x_{44} may increase in this iteration (step 4, case a). Therefore we remove (4, 4) from B_2 and introduce (3, 2) since $\min_{(i,j)\in\Pi_2} t_{ij} = t_{32} = 11$. Thus $B_3 = B_2 - \{(4, 4)\} + \{(3, 2)\}$.

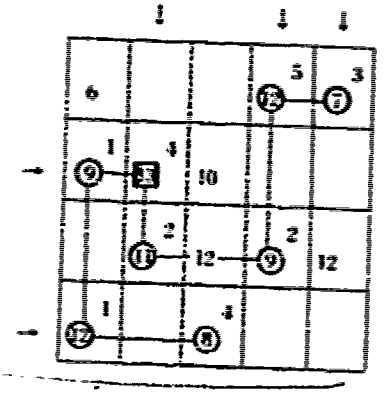
$$X(B_3) = X(B_2), \text{ and } \overline{\Pi}_3 = \overline{\Pi}_2.$$



We repeat the procedure from steps 2–4 and obtain $X(B_4)$.



where $\Pi_4 = \Pi_3 + \{(2, 4), (2, 5), (3, 1), (4, 2)\}$ and further $X(B_5)$ where $\Pi_5 = \Pi_4$.



Determine the corresponding set Ψ. Since $\Psi\bar{\Pi}_2$ is empty $X(B_2)$ is an optimal solution.

Note that each $\bar{\Pi}_4$ solution is optimal. Therefore $X(B_4)$ is also an optimal solution as well as the two solutions given below:

			7	1
1	4			
	2			2
1		4		

			6	2
1	4			
	2		1	1
1		4		

Here the second optimal solution is not basic.

4. Appendix

Proof of Theorem 1 from [5] (pp. 346–347)

THEOREM 1: A feasible [basic] solution is optimal if, and only if, it is locally optimal.*

Remark 4: We will use in the proof which follows notations and numbers of formulae as in [5]. The only change is that we replace "stars" of [5] by numbers and we also replace his C by z.

PROOF: We will prove only that a local optimal solution is optimal (since the second part of the theorem is obvious). Let $X_1=\{x_{ij}^1\}$ be a feasible basic solution which is locally optimal and suppose on the contrary that there exists feasible solution (basic or not) $X_2=\{x_{ij}^2\}$ which is better than X_1.

Introduce a $m \times n$ cost matrix $C=\{c_{ij}\}$, where†

$$c_{ij}=\begin{cases} d=\sum_i a_i+1=\sum_j b_j+1 & \text{if } t_{ij}>t_i \\ 1 & \text{if } t_{ij}=t_i \\ 0 & \text{if } t_{ij}<t_i \end{cases} \tag{12}$$

Considering X_1 and X_2 as solutions of a "cost" transportation problem with a cost function $z=\sum c_{ij}t_{ij}$, we have

$$z_1=\sum_{i,j} c_{ij}x_{ij}^1=p_1>0 \tag{13}$$

and

---

*I inserted the term "basic" which is in accordance with the intention of the author who used this fact in the proof when he introduced solution X_1 adjacent to X_0.

†One should stress that formula (12) as given in [5] is not sufficient to prove that $z_1 \le z_2$, since then $z_0=p_0$, but p_1 may be greater than $m+n-1$ (the author of [5] claims that $z_1 \le m+n-1$ which helped him to prove that $z_2 \le z_1$) because p_1 is a sum of some x_{ij}. One should note that equation $\sum c_{ij}x_{ij}^1=t_1p_1$ (see (14) in [5]) is not correct since the left side is either 0 or p_1 while the right side is $t_1p_1>0$ (t_1p_1 may be zero only if $t_1=\max_{(i,j)\in\chi_1} z_{ij}=0$).

Also relation (15) in [5] is a strict inequality that χ_1 is not optimal and therefore exists a cheaper adjacent basic solution.

(11) $$z_2 = \sum_{i,j} c_{ij} x_{ij}^2 .$$

Since X_2 is better than X_1 (which means that either $t_1 > t_2$ or if $t_1 = t_2$ then

$$p_1 = \sum_{i,j \in I_1} x_{ij}^1 > \sum_{i,j \in I_2} x_{ij}^2 = p_2),$$

we have

$$z_2 = \begin{cases} 0 \text{ if } t_1 > t_2 \\ p_2 \text{ if } t_1 = t_2. \end{cases}$$

For either case $z_2 < z_1$ which implies that X_1 is not an optimal solution of the cost transportation problem with the cost matrix C defined by (12).

This means that there exists an adjacent to X_1 basic feasible solution X_2 for which

$$z_2 = \sum_{i,j} c_{ij} x_{ij}^2 < z_1.$$

Consider t_2. Inequality $t_2 > t_1$ will imply $z_2 \geq d$ which is impossible since z_2 is less than z_1 and $z_1 = p_1 < d$ (p is a sum of some x_{ij} while d is greater than the sum of all x_{ij}).

We conclude, therefore, that $t_2 \leq t_1$. There are two cases.

a) Let $t_2 = t_1$. This, together with $z_2 < z_1$ implies $p_2 < p_1$.

b) Let $t_2 < t_1$. Then $p_2 = 0$.

This in turn implies that X_2 is a better solution than X_1 which contradicts the assumption that X_1 is locally optimal. This completes the proof of theorem 1.

REFERENCES

[1] A. R. Barsow, *What is Linear Programming* (Moscow, 1959), pp. 90–101 (in Russian).

[2] R. S. Garfinkel and M. R. Rao, "The Bottleneck Transportation Problem," Nav. Res. Log. Quart., **18**, 465–484 (1971).

[3] W. Grabowski, "Problem of Transportation in Minimum Time," Bull. Acad. Polon. Sci., **12**, 107–108 (1964).

[4] W. Grabowski, "Transportation Problem with Minimization of Time," Przeglad Statystyczny, **11**, 333–359 (1964) (in Polish).

[5] P. L. Hammer, "Time Minimizing Transportation Problems," Nav. Res. Log. Quart., **16**, 345–357 (1969).

[6] A. Janicki, "The Time Transportation Problem," M. S. Thesis, April 1969, University of Wroclaw, Poland (in Polish).

[7] E. P. Niestierow, *Transportation Problems in Linear Programming* (Moscow, 1962), pp. 72–80 (in Russian).

[8] W. Szwarc, "Das Transportzeitproblem," Mathematik und Kybernetic in der Oekonomie, Aca-

demie Verlag Berlin **19**, 72–78 (1965) (in German). (Proceedings of the International Conference of the Applications of Mathematics and Cybernetics to Economies, Berlin, October 1964).

[9] W. Szware, The Time Transportation Problem, Zastosowania Matematyki **8**, 231–242 (1966).

[10] W. Szware, "The Time Transportation Problem," Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, Technical Rept No. 201 (Feb. 1970).

[11] S. I. Zukhovitskiy and L. I. Avdeyeva, *Linear and Convex Programming* (W. B. Saunders Co., Philadelphia and London 1966) pp. 160–169.

COMMUNICATION ON "THE BOTTLENECK TRANSPORTATION PROBLEM" AND "SOME REMARKS ON THE TIME TRANSPORTATION PROBLEM"

§ 1. Along with sending me the manuscript of their remarkable paper [3] on time minimizing transportation problems, Professors R. S. Garfinkel and M. R. Rao have kindly called my attention [2] to the fact that the algorithm I have given in [4] for the solution of the same problem must contain an error, since they have found examples where it does not lead to an optimal solution. The counter example of [2] will be given below.

Unfortunately, the procedure (though it can be "easily corrected" [2]) is not correct when (using the notations of [4]), $|N| > 1$. A duly corrected version of the algorithm will be given here.

§ 2. The counterexample of Garfinkel and Rao is the following. Consider the time minimizing transportation problem with

supplies: 37. 22. 31. 14
demands: 15. 20. 15. 24. 20. 10
and with

$$\{t_{ij}\} = \begin{array}{|c|c|c|c|c|c|} \hline 25 & 20 & 20 & \times & 30 & \times \\ \hline \times & \times & 45 & 30 & \times & \times \\ \hline \times & \times & \times & 45 & 45 & \times \\ \hline \times & \times & \times & \times & 30 & 25 \\ \hline \end{array}$$

An initial basic feasible solution is

15	20	2			
		13	9		
			15	16	
				4	10

Here $t^* = 45$ and $N = \{(2.3), (3.4), (3.5)\}$. The algorithm of [4] would recommend the reduction of $x_{1,3}$, i.e. the introduction of the cell (4.5) into the basis, leading to the following new basic feasible solution

15	20			2	
		15	7		
			17	14	
				4	10

which, however, is worse than the first one.

§3. What was wrong with my algorithm given in [4]? The procedure enabled to reduce the amount of an x_{hk} with $(h, k) \in N$, overlooking the fact that it might happen that by introducing a new element x_{ij} into the basis, as a side effect, the values of some other $x_{h'k'}$ (with $(h', k') \in N$) may increase (in the above example $x_{2,3}$ and $x_{3,4}$). Hence, a correct procedure must make sure that $t_{ij} \leq t^*$ and that

$$\sum_{(hk) \in N} x_{hk} + \epsilon_{ij} x_{ij} \text{ where } \epsilon_{ij} = 1 \text{ if } t_{ij} = t^*, \text{ and } \epsilon_{ij} = 0 \text{ if } t_{ij} < t^*)$$

decrease step by step.

§4. *Correction.* Let us define for an arbitrary $(i, j) \notin M$, and an arbitrary $(h, k) \notin N$, the values $u_{h,k}(i)$ and $v_{h,k}(j)$ as in (17), (18) of [4] (where these values were denoted by $u(i)$ and $v(j)$, respectively). Let us further put

$$z_{h,k}(i,j) = u_{h,k}(i) + v_{h,k}(j) - 1.$$

The following two Lemmas have been proved in [1]:

Lemma A. By introducing (i, j) into the basis, the value x_{hk} decreases if and only if $z_{h,k}(i, j) = -1$.

Lemma B. By introducing (i, j) into the basis, the value x_{hk} increases if and only if $z_{h,k}(i, j) = 1$.

If we denote now by S the set of those $(i, j) \notin M$, the introduction of which into the basis leads to a better solution, we arrive at the following.

THEOREM. If for every $(i, j) \notin M$, we put

$$\epsilon_{ij} = \begin{cases} 1 \text{ if } t_{ij} = t^* \\ 0 \text{ if } t_{ij} < t^* \\ m + n \text{ if } t_{ij} > t^* \end{cases}$$

and

$$Z(i, j) = \epsilon_{ij} + \sum_{(h,k) \in N} z_{h,k}(i, j).$$

then

$$S = \{(i, j) \mid (i, j) \in M, Z(i, j) < 0\}.$$

Hence, step 2) of the procedure given in [4] will have to be the following:

2-1) Determine S (by the above Theorem);

2-2) Determine $(i, j) \in S$ for which the absolute value of $Z(i, j)$ is maximal;

2-3) Introduce (i, j) into the basis (as in the common transportation problem).

Remark. If $|N| = 1$, the algorithm is identical with that of [4].

§5. As an example, consider the first solution of the Garfinkel-Rao counter example. Here,

$$N = \{(2, 3), (3, 4), (3, 5)\},$$

and, we can easily find the values of the $Z_{h,k}(i, j)$ for $(h, k) \in N$, $(i, j) \notin M$ (in place of the elements of M the corresponding level L_r containing it was introduced):

Tableau of $z_{2,3}(i, j)$'s

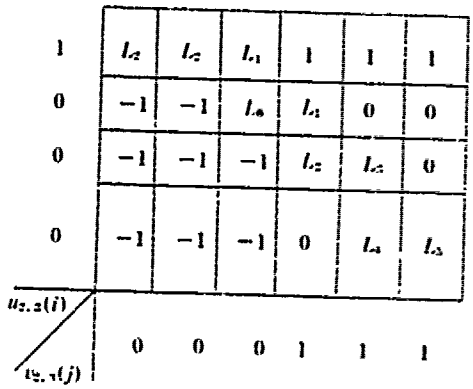


Tableau of $z_{3,4}(i, j)$'s

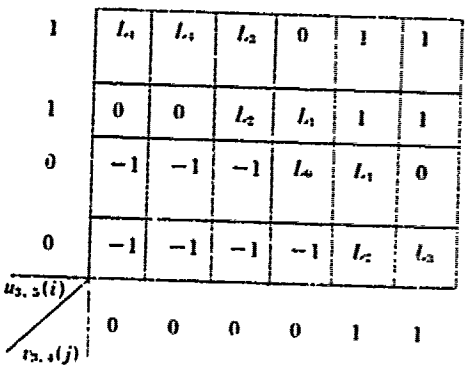
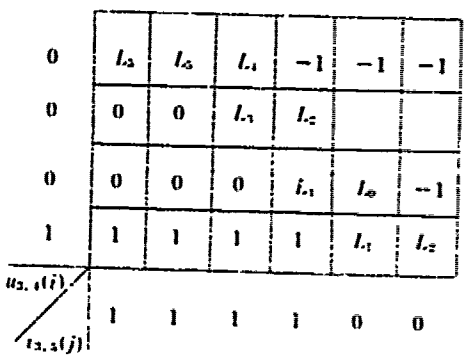


Tableau of $z_{3,5}(i, j)$'s



Finally, the tableau of the ϵ_{ij}'s for $(i, j) \in M$ is

			10	0	10
10	10			10	10
10	10	10			10
10	10	10	10		

Hence, the tableau of the $Z(i, j)$'s for $(i, j) \in M$, is

			10	1	11
9	9			10	10
8	8	8			9
9	9	9	10		

Hence all $Z(i, j)$ are positive, showing that the first solution is optimal.

§ 6. The Editor of NRLQ has kindly brought to my attention the manuscript of [5].

Although I cannot agree with numerous statements of [5], I would like to stress its positive aspect.

I am happy to learn of the contributions of L. I. Avdeyeva, A. S. Barsow, W. Grabowsky, A. Janicki, E. P. Niesterow, W. Szwarc and S. I. Zukhovitskiy to the time minimizing transportation problem.

§ 7. Finally, I would like to express my appreciation to Professors Garfinkel and Rao for having called my attention to the error contained in my paper, and to the Editor of NRLQ for having informed me about Szwarc's paper and for the kind publication of this Letter.

REFERENCES

[1] E. Balas and P. L. Hammer (Ivanescu): On the Transportation Problem. *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, Part I in **4**, 1962, pp. 98–116; part II in **4**, 1962, pp. 131–160.

[2] R. S. Garfinkel and M. R. Rao: Private Communication. May 1970.

[3] R. S. Garfinkel and M. S. Rao: The Bottleneck Transportation Problem. Nav. Res. Log. Quart. **18**, pp. 465–472, 1971.

[4] P. L. Hammer: Time Minimizing Transportation Problems. Nav. Res. Log. Quart. **16**, pp. 345–357, 1969.

[5] W. Szwarc: Some Remarks on the Time Transportation Problem. Nav. Res. Log. Quart. **18**, pp. 473–485, 1971.

Peter L. Hammer
Centre de recherches mathématiques
Université de Montréal
Montréal, Canada

August 1971.

INTEGER POINTS ON THE GOMORY FRACTIONAL CUT (HYPERPLANE)

Harvey M. Salkin and Patrice Breining

Department of Operations Research
Case Western Reserve University
Cleveland, Ohio

ABSTRACT

In this paper we show that the Gomory fractional cut (hyperplane) for the integer program is either void of integer points or contains an infinite number of them. The conditions for each case are presented. Also, we derive a stronger cut from the hyperplane which does not intersect integer points.

In Reference [1] Gomory develops the well known fractional cut (1) (row indices are omitted) as part of a classical cutting plane.

$$s=-f_0+\sum_{j=1}^{n} f_j x_{J(j)} \geq 0 \quad (0<f_0<1,\ 0\leq f_j<1). \tag{1}$$

algorithm for the integer program. The inequality (1), implied by the constraints of the integer program, is obtained from a source row.

$$x=a_0+\sum_{j=1}^{n} a_j(-x_{J(j)}) \qquad (a_0>0 \text{ and not integral}), \tag{2}$$

where a_j is some integer linear combination of the coefficients in column j of the current simplex tableau, and $J(j)$ is the jth index ($j=1, \ldots, n$) in the set of indices J corresponding to the current nonbasic variables. Also, $f_j=a_j-[a_j]$, where $[y]$ is the largest integer smaller than or equal to y.

RESULTS

A well known result is that every a_j in (2) may be written as $I'_j/|B|$, where I'_j is an integer and $|B|$ is the determinant of the current basis. Also, as indicated in Reference [1], we have

$$f_j=a_j-[a_j]=\frac{I'_j}{|B|}-[a_j]=\frac{I'_j-[a_j]|B|}{|B|}=\frac{I_j}{|B|} \quad (j=0, 1, \ldots, n)$$

where I_j ($j=1, \ldots, n$) is a nonnegative integer, I_0 is a positive integer, and all I_j are smaller than $|B|$. (This follows since the original tableau is assumed to be integral. Hence, $|B|$ is an integer. Also, it is the product of the pivot elements which is always positive.) Consequently, the inequality (hyperplane) (1) may be written as

$$s=\frac{-I_0}{|B|}+\sum_{j=1}^{n}\frac{I_j}{|B|} x_{J(j)} \geq (=)\ 0. \tag{3}$$

We are now ready to present some useful properties.

THEOREM: Consider the hyperplane $\frac{I_o}{|B|}=\sum_{j=1}^{n}\frac{I_j}{|B|}xJ(j)$ corresponding to a generated inequality (1). Then the hyperplane passes through at least one integer point (not necessarily a feasible solution to the integer program) if, and only if, the greatest common divisor (gcd) of $I_1, \ldots, I_n$ divides I_o. Moreover, if it contains one integer point it contains an infinite number of them.

PROOF: Suppose $(\lambda_1, \ldots, \lambda_r)$ is an integer point. Then, for it to be on the hyperplane we must have $\frac{I_o}{|B|}=\sum_{j=1}^{n}\frac{I_j}{|B|}\lambda_j$, which means $I_o=\sum_{j=1}^{n}I_j\lambda_j$. The entire assertion then follows from elementary results in Number Theory (see, Reference [5], Theorem 4–1, p. 169 and Theorem 4–3, p. 176).

To illustrate, consider example 2 of Reference [1]. The integer program is

$$\begin{aligned} &\text{maximize} && 3x_1 - x_2 = x_o, \\ &\text{subject to} && 3x_1 - 2x_2 \leq 3 \\ & && -5x_1 - 4x_2 \leq -10 \\ & && 2x_1 + x_2 \leq 5, \\ &\text{and} && x_1, x_2 \geq 0, \text{ integer} \end{aligned}$$

With nonnegative slacks x_3, x_4, and x_5, the (first) optimal simplex tableau is

	1	$-x_3$	$-x_5$
x_o	30/7	5/7	3/7
x_1	13/7	1/7	2/7
x_2	9/7	−2/7	3/7
x_3	0	−1	0
x_4	31/7	−3/7	22/7
x_5	0	0	−1

$$B=\begin{pmatrix} 1 & 0 & 0 \\ -3 & -3 & 2 \\ -5 & -2 & -1 \end{pmatrix}$$

$$|B| = 7$$

Suppose $2x_1$ is the source row. That is, the cut is generated from $2x_1 = 26/7 - 2/7x_3 - 4/7x_5$. Then the derived inequality is

$$s = -5/7 + 2/7x_3 + 4/7x_5 \geq 0.$$

Now, the gcd of 2 and 4 is 2 which does not divide 5. Hence, there cannot exist an integer point satisfying $s = 0$. To explicitly see this, transform the hyperplane to (x_1, x_2) space. The result is the hyperplane $3 - 2x_1 = 0$ which, of course, does not intersect any integer point (see Figure 1).

Suppose now x_1 is used as the source row. Then the generated inequality would be

$$s' = -6/7 + 1/7x_3 + 2/7x_5 \geq 0.$$

Here, the gcd of 1 and 2 is 1 which divides 6. Hence, the hyperplane $s' = 0$ passes through integer

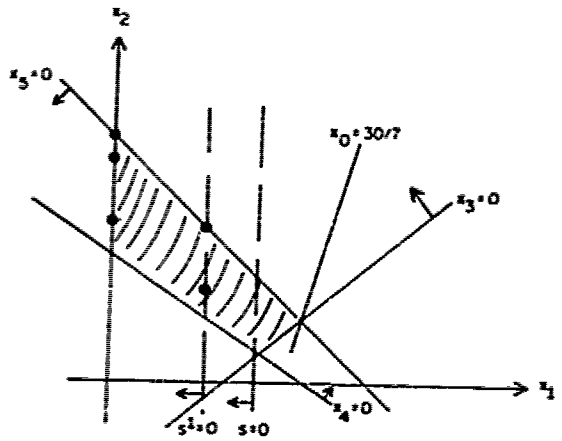


FIGURE 1.

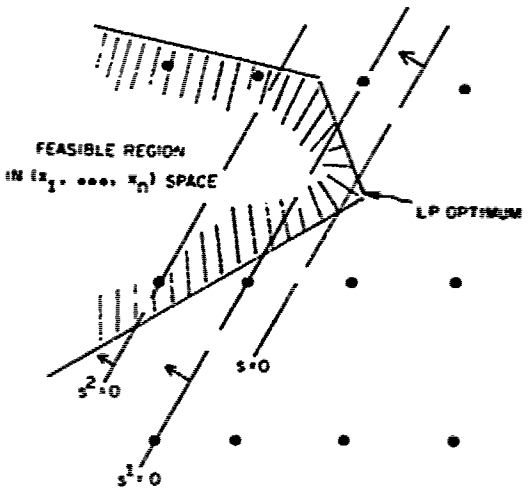


FIGURE 2.

points. To see this write $s^1=0$ in terms of x_1 and x_2. This yields the hyperplane $2-2x_1=0$ in (x_1, x_2) space. It intersects an infinite number of integer coordinates (see Figure 1).

IMPROVING THE CUT

To complete the discussion consider the case where the hyperplane $s=0$ is void of integer points. Then the inequality (3) can be improved if we can push its hyperplane into the feasible region until it intersects the first integer point (on Figure 2, the inequality $s^1 \geq 0$ is stronger than $s \geq 0$). To do this, rewrite inequality (3) as

$$x=-\frac{l_0}{|B|}+\sum_{j=1}^{n}\frac{l_j}{|B|}x_{j(i)} \geq 0. \tag{3}$$

Since there is no integer point satisfying $s=0$, we know that the gcd of $l_1, \ldots, l_n$, say d, does not divide l_0.

We can also write the inequality $s \geq 0$ in terms of the original nonbasic variables $x_1, \ldots, x_n$.

The result (see, Reference [1]) is the all integer inequality

$$s = n_0 + \sum_{j=1}^{n} n_j x_j \geq 0. \tag{4}$$

Now, there is a 1-to-1 correspondence between the points of the feasible region in $(x_{J(1)}, \ldots, x_{J(n)})$ space (which is contained in the first quadrant) and those in $(x_1, \ldots, x_n)$ space (defined by the original constraints). Further, the hyperplane (3) intersects the $x_{J(j)}$ axis at the point $\frac{I_0}{I_j}$ (see Figure 3). Therefore, to "push" the inequality $s \geq 0$ (parallel to itself) into the feasible region, or equivalently, to derive a stronger inequality, we must *increase* I_0. Looking at (4), we can change n_0 by 1 without cutting off any integer point: but increasing $\frac{I_0}{|B|}$ (or f_0) by 1 is the same as changing n_0 by 1. Thus, we can increase $\frac{I_0}{|B|}$ by 1 without cutting off any integer point: or equivalently, we can increase I_0 by $|B|$. This yields the stronger inequality,

$$s^1 = -\frac{\bar{I}_0}{|B|} + \sum_{j=1}^{n} \frac{I_j}{|B|} x_{J(j)} \geq 0, \tag{5}$$

where $\bar{I}_0 = I_0 + |B|$. (Note that I_j cannot be increased by any integer amount.)

Now the same line of reasoning can be applied to the new inequality (hyperplane) (5): that is, if d (which has not changed) divides $I_0 + |B|$, we have the desired constraint. Otherwise, we increase $\bar{I}_0 = I_0 + |B|$ by $|B|$ and retest d. In effect, the process is repeated until we find the smallest positive integer K, such that d divides $I_0 + K|B|$. To clarify the procedure we improve the first inequality in the example presented earlier. We used the source row

$$2x_1 = 26/7 + 2/7(-x_2) + 4/7(-x_5).$$

The generated inequality (hyperplane) was ($|B| = 7$)

$$s = -5/7 - 2/7(-x_2) - 4/7(-x_5) \geq (=)0. \tag{3}'$$

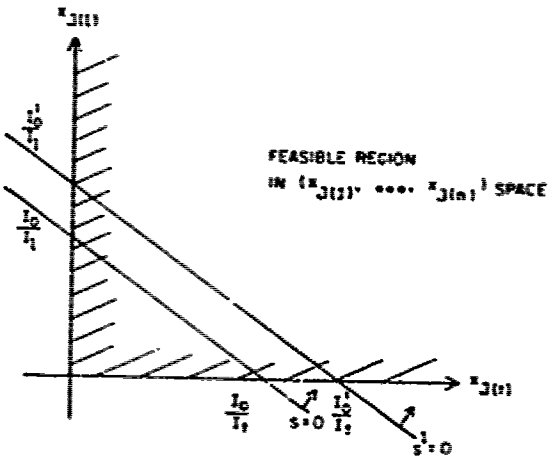


FIGURE 3.

Or, in terms of the original nonbasic variables x_1 and x_2

(4)' $$s=3-2x_1 \geq 0.$$

Now, the gcd of 2 and 4 is 2, which does not divide 5. Hence, add 1 to 5/7 in (3)' and the new cut is

(5)' $$s^1=-12/7-2/7(-x_2)-4/7(-x_3) \geq 0.$$

Its hyperplane intersects integer points (since 2 divides 12). Of course, increasing 5/7 by 1 in (3)' amounted to reducing 3 by 1 in (4)'. Or, in terms of x_1 and x_2, (4)' is

(5)' $$s^1=2-2x_1 \geq 0.$$

which clearly intersects an infinite number of integer points. (see Figure 1).

COMMENTS

1. Gomory [2] has shown that the integer program: max $c_Bx_B+c_Nx_N$, subject to $Bx_B+Nx_N=b$, $x_B, x_N \geq 0$ and integer, where B is a (current) optimal linear programming basis, without the condition $x_B \geq 0$, is equivalent to the group minimization problem: minimize $(c_N-c_BB^{-1}N)x_N$, subject to $\sum_{j=1}^{n} g_jx_{N(j)}=g_0$, $x_N=(x_{N(1)}, \ldots, x_{N(n)}) \geq 0$ and integer, where the vector g_j, $j=0, 1, \ldots, n$ is an element of the factor group $\{B^{-1}\}/\{I\}$ and is the image of the nonbasic column $a_j (j \neq 0)$ or $b (j=0)$. ($\{A\}$ is the set of vectors which can be written as an integer linear combination of the columns of A and I is an identity matrix.) As Hu indicates in reference 4 the matrix $g=(g_0, g_1, \ldots, g_n)$ can be found by transforming each row of $B^{-1}(b, N)$ to the row vector representing the coefficients of the corresponding Gomory cut, multiplying this matrix by $|B|$, and premultiplying the resulting (integer) matrix by a unimodular (row transformation) matrix Q^{-1}. That is, each equality in $\sum_{j=1}^{n} g_jx_{N(j)}=g_0$ has the form

$$Q^{-1}(I_1, \ldots, I_n)x_N=Q^{-1}I_0.$$

Equivalently, $\sum_{j=1}^{n} I_jx_{N(j)}=I_0$. Thus, it follows that if a Gomory cut passes through integer points then its corresponding group equation also contains integer solutions and vice versa.

Another interesting thing is that the hyperplanes defining the boundaries of the feasible region of the group problem (referred to as "faces") give the strongest cuts that can be generated from the current tableau. (See Reference [4].) Thus, the faces must correspond to cuts whose hyperplanes contain integer points.

2. Up to this time we have not mentioned integer points *inside* the feasible region. A generated inequality (hyperplane), such as (3), could be *most* improved if we could replace it by the one whose hyperplane passes through the first integer solution. (In Figure 2 the hyperplane $s^2=0$ defines the strongest cut from $s=0$.) The difficulty is, however, to determine whether the hyperplane passes through an integer solution and, if not, to obtain one that does. Precise conditions, as before, are not available. Nevertheless, a procedure might be to first obtain the hyperplane which passes through an integer point. Then, test all integer points on it "near" the feasible region for solutions. If an integer solution cannot be found increase f_0 by 1 and repeat the search. The problem with this approach is that usually there are many integer points near the feasible region on the hyperplane. This is especially

true when $n-1$, the dimension of the hyperplane, is large. Thus, even if a systematic enumeration scheme could be developed it would almost surely be computationally unworthy.

REFERENCES

[1] Gomory, R. E., "An Algorithm for Integer Solutions to Linear Programs," Princeton IBM Math. Res. Report (Nov. 1958); also in R. L. Graves and P. Wolfe (eds.), *Recent Advances in Mathematical Programming* (McGraw Hill, New York, 1963), pp. 269–302.

[2] Gomory, R. E., "Some Polyhedra Related to Combinatorial Problems," *Linear Algebra and Its Applications* (American Elsevier Publishing Company, Inc., New York, N.Y., 1969), Vol. 2, pp. 451–558.

[3] Gomory, R. E., "Properties of a Class of Integer Polyhedra," Chap. 16 in *Integer and Nonlinear Programming*, edited by J. Abadie (North-Holland/American Elsevier, New York, N.Y., 1970).

[4] Hu, T. C., *Integer Programming and Network Flows* (Addison Wesley, Reading, Mass., 1969).

[5] Saaty, T. L., *Optimization in Integers and Related Extremal Problems* (McGraw Hill, New York, N.Y., 1970).

DETERMINING THE MOST VITAL LINK IN A FLOW NETWORK*

S. H. Lubore

The MITRE Corporation, McLean, Virginia

H. D. Ratliff

The University of Florida, Gainesville, Florida

G. T. Sicilia

*Research Analysis Corporation, McLean, Virginia***

ABSTRACT

The most vital link in a single commodity flow network is that arc whose removal results in the greatest reduction in the value of the maximal flow in the network between a source node and a sink node. This paper develops an iterative labeling algorithm to determine the most vital link in the network. A necessary condition for an arc to be the most vital link is established and is employed to decrease the number of arcs which must be considered.

INTRODUCTION

The problem of removing arcs and nodes from a network is a part of network theory that has many important and useful applications. One application would be a conflict situation where there is a logistics or communications network under attack. A defender or user of the system must know which arcs are most vital to him so that he can reinforce them against attack; while the attacker, naturally, wants to destroy those arcs whose destruction would most affect the efficiency of the system. Another application would be in helping the managers of a highway system or a transportation network to determine the effect of closing various links for repair, etc.

The problem addressed by this paper is concerned with finding the most vital link in a single commodity flow network $[V;A]$ (directed, undirected, or mixed). An arc $(x,y)\in A$ is declared to be the most vital link if its value $r(x,y)$ is at least as large as the value of every other arc in the network. The value of arc (x,y) is defined as the difference in the maximal flow values in networks $[V; A]$ and $[V; \{A-(x,y)\}]$ between some source node and some sink node. Thus, $r(x,y)$ reflects the reduction in the maximal flow attainable if arc (x,y) is removed from the network.

Wollmer [3] has developed an algorithm for determining the most vital network link. This algorithm has been employed by Durbin [1] to determine the single most critical link in a highway system.

The following sections of this paper briefly present Wollmer's method for finding the most vital link in a network and develop an improved algorithm for solving the most vital link problem. An example using the improved method is included.

WOLLMER'S ALGORITHM

Wollmer's algorithm for finding the most vital link in a network follows as a consequence of the

*Presented at the 38th National ORSA Meeting, Detroit, Michigan, October 28-30, 1970.

**This work was performed while the author was at The Mitre Corporation.

following theorem. The proof of this theorem is given in Reference [3].

Theorem (Wollmer):

Suppose the network $[N; A]$ has a maximal flow value of $v(F^*)$, while the network $[N; \{A-(x, y)\}]$ has a maximal flow value of $v(F^*_{xy})$. Then, every maximal flow pattern of $[N; A]$ has at least $v(F^*) - v(F^*_{xy})$ units of flow through the arc (x, y). Moreover, there is a maximal flow pattern of $[N; A]$ which has exactly $v(F^*) - v(F^*_{xy})$ units of flow over the arc (x, y).

As Wollmer points out, "the above theorem reduces the problem to one of finding that link whose minimal flow among all maximal flow patterns is greatest." Wollmer's iterative procedure for finding this link is as follows:

STEP 0:

Find a maximal flow pattern, F^*, in the network $[N; A]$ and let $f^*(x, y)$ be the corresponding flow in each arc $(x, y) \in A$.

Set the "least flow" of each arc (x, y), equal to $f^*(x, y)$.

Let $f^*(p, q) = \max_{(x, y) \in A} f^*(x, y)$ and go to STEP 1.

STEP 1: Solve a maximal flow problem for the network $[N; \{A-(p, q)\}]$. Let the corresponding maximal flow pattern be denoted as F^*_{pq} and go to STEP 2.

STEP 2:

(a) Set the capacity $c(p, q)$ of arc (p, q) equal to $v(F^*) - v(F^*_{pq})$ and solve a maximal flow problem for this network (i.e., the network $[N; A]$ with $c(p, q) = v(F^*) - v(F^*_{pq})$). Call the corresponding maximal flow pattern F'. (Note F' is also a maximal flow pattern of $[N; A]$.)

(b) Next, compare the least flow of each arc (x, y) with $f'(x, y)$ and if $f'(x, y) <$ least flow of arc (x, y), replace the least flow of (x, y) with $f'(x, y)$. Reset $c(p, q)$ to its original value and go to STEP 3.

STEP 3: Let $U = \max_{(x, y) \in A}$ {least flow of arc (x, y)}.

(a) If $U \leq v(F^*) - v(F^*_{pq})$ terminate; (p, q) is a most vital link;

(b) If $U > v(F^*) - v(F^*_{pq})$: find an arc (p, q), such that the least flow of (p, q) equals U, and go to STEP 1.

AN IMPROVED ALGORITHM

Wollmer's algorithm considers each arc as a candidate for the most vital link; however, a necessary condition is employed in the improved algorithm which reduces the number of arcs that must be considered explicitly as candidates.

THEOREM 1: A necessary condition for an arc (a, b) to be a most vital link is that for any maximal flow pattern in the network $[N; A]$, the flow in arc (a, b) is at least as great as the flow over every arc in a minimal cut.

The following lemma will be useful in the proof of Theorem 1.

LEMMA 1: If $(X, \bar{X})$ is a minimal cut containing at least two arcs in a network $[N; A]$ and if arc (x, y) is in $(X, \bar{X})$, then $(X, \bar{X}) - (x, y)$ is a minimal cut in the network $[N; \{A-(x, y)\}]$.

PROOF OF LEMMA 1: Suppose that $(Y, \bar{Y})$ is a minimal cut in $[N; \{A-(x, y)\}]$ and that $C(Y, \bar{Y}) < C(X, \bar{X}) - c(x, y)$. Note that $(Y, \bar{Y}) \cup (x, y)$ has to be a disconnecting set for $[N; A]$ and that $C(Y, \bar{Y}) + c(x, y) < C(X, \bar{X})$; but $(X, \bar{X})$ is a minimal cut of $[N; A]$ and $(Y, \bar{Y}) \cup (x, y)$ is a disconnecting

set of $[N; A]$. Thus, it must be true that $C(Y, \bar{Y}) = C(X, \bar{X}) - c(x, y)$ *and, hence,* $(X, \bar{X}) - (x, y)$ is a minimal disconnecting set and thus a minimal cut in $[N; \{A - (x, y)\}]$.

PROOF OF THEOREM 1: Let $(X, \bar{X})$ be a minimal cut in $[N; A]$ and note that by hypothesis arc (a, b) is a most vital link. Assume that $f^*(a, b) < f^*(p, q) = \max_{x,y \in (X, \bar{X})} f^*(x,y)$ for some maximal flow pattern F^* defined in $[N; A]$. It will be shown that this assumption leads to a contradiction.

(1) By Lemma 1: $v(F^*_{pq}) = v(F^*) - f^*(p, q)$: by assumption $f^*(a, b) < f^*(p, q)$: therefore, $v(F^*_{pq}) = v(F^*) - f^*(p, q) < v(F^*) - f^*(a, b)$.

(2) Further, there exists a flow pattern in $[N; \{A - (a, b)\}]$ with the value $v(F^*_{ab}) = v(F^*) - f^*(a, b)$, and hence, the maximal flow value, $v(F^*_{ab})$, in $[N; \{A - (a, b)\}]$, must satisfy: $v(F^*_{ab}) \geq v(F^*) - f^*(a, b)$.

(3) Conditions (1) and (2) imply that $v(F^*_{ab}) > v(F^*_{pq})$: but this leads to a contradiction since by assumption (a, b) is a most vital link and by the definition of a most vital link it follows that:

$$v(F^*_{ab}) \leq v(F^*_{xy}), \text{ for all } (x, y) \in A. \qquad \textit{Q.E.D.}$$

Theorem 1 guarantees that those arcs whose flow is less than the largest flow through an arc in a minimal cut for some maximal flow pattern in $[N; A]$ need not be considered as candidates for the most vital link.

A LABELING SCHEME

A labeling scheme is employed in the algorithm for finding a most vital link from the set of candidate arcs. At the outset of the labeling, it is assumed that there is a maximal flow pattern, and minimal cut defined for $[N; A]$. Suppose that arc (a, b) is an arc from the candidate set (this set is made up of those arcs that satisfy the necessary condition of Theorem 1). Next, an attempt is made to label from node a to node b without using the arc (a, b). Initially node a is labeled and all other nodes are unlabeled. The labeling scheme systematically searches for a flow path from node a to node b which does not include arc (a, b) such that flow may be diverted from arc (a, b) over this path. If node b is labeled (breakthrough), such a path exists. The labeling (and backtracking) rules to be used are similar to those employed by Ford and Fulkerson [2] in their algorithm for the solution of the maximal flow problem. The labeling and flow changing rules are:

Let node a be labeled with $(-, \epsilon(a) = \infty)$. At a general step, suppose the node x is labeled $(z^{\pm}, \epsilon(x))$ and that the nodes x and y are connected by some arc. Then, node y may be labeled if either of the following situations occur:

(a) $f(y,x) > 0$: then node y is labeled $(x^-, \epsilon(y))$ where $\epsilon(y) = \min(\epsilon(x), f(y,x))$: or

(b) $f(x,y) < c(x,y)$: then node y is labeled $(x^+, \epsilon(y))$ where $\epsilon(y) = \min(\epsilon(x), c(x,y) - f(x,y))$.

The labeling process is continued until either node b is labeled (breakthrough), or node b is unlabeled in which case no more labeling is possible (non-breakthrough). If breakthrough occurs, node b must have a label of the form $(q^{\pm}, \epsilon(b))$ for some node q. Likewise, node q has a label $(r^{\pm}, \epsilon(q))$ and node r has a label $(p^{\pm}, \epsilon(r))$, etc. Thus a series of nodes (and a path of arcs) starting with node b and ending with node a is defined.

Let $\epsilon = \min[\epsilon(b), f(a,b)]$. For each arc (x,y) on this path, change the flow as follows:

(a) If node y has a label $(x^+, \epsilon(y))$, replace $f(x,y)$ by $f(x,y) + \epsilon$;

(b) If node y has a label $(x^-, \epsilon(y))$, replace $f(y,x)$ by $f(y,x) - \epsilon$. Finally, decrease the flow in arc (a,b) by ϵ units.

As Ford and Fulkerson have shown the labeling scheme must end in one of two mentioned states: breakthrough is achieved or breakthrough is not achieved. The following results indicate what can be deduced when either of these states occur at the end of the labeling process.

THEOREM 2:

Let F^* be a maximal flow pattern in the network $[N;A]$. Let (a,b) be an arc in $[N;A]$ and use the labeling scheme to label from node a to node b without using arc (a,b). If breakthrough occurs, the value of arc (a,b) is less than or equal to $f^*(a,b)-\epsilon$. If non-breakthrough occurs, the value of arc (a,b) is equal to $f^*(a,b)$.

PROOF: If breakthrough occurs an alternate maximal flow pattern is found by making an ϵ change of flow in the path established by the labeling and decreasing the flow in (a,b) by ϵ. Then $v(F^*_{ab}) \geq v(F^*)-f^*(a,b)+\epsilon$ and since $v(a,b)=v(F^*)-v(F^*_{ab}) \leq v(F^*)-[v(F^*)-f^*(a,b)+\epsilon]$ and $\epsilon>0$, it follows that $v(a,b) \leq f^*(a,b)-\epsilon < f^*(a,b)$.

Assume that non-breakthrough occurs, then by the nature of the labeling rules, no maximal flow pattern exists in $[N;A]$ which has less than $f^*(a,b)$ units of flow over arc (a,b) and, hence, by Wollmer's Theorem $v(a,b)=f^*(a,b)$. *Q.E.D.*

Utilizing these results, the following algorithm can be employed to locate a most vital link in a network:

STEP 0:

(a) Find a maximal flow pattern, F^*, in the network $[N;A]$ and let $(X,\bar{X})$ be a minimal cut.* Let $U^*=\max_{(x,y)\in(X,\bar{X})} c(x,y)$ or, alternately, $U^*=\max_{(x,y)\in(X,\bar{X})} F^*(x,y)$.

(b) Note those arcs $(x,y)\in A$ for which $f^*(x,y) \geq U^*$ and store these arcs in a list; these arcs form the candidate set, S. For each arc in this set define an upper bound as $U(x,y)=f^*(x,y)$.

STEP 1:

Let $U(a,b)=\max_{(x,y)\in S} U(x,y)$ and set $f(x,y)=f^*(x,y)$ for all arcs in $[N;A]$.

STEP 2:

Use the labeling rules to label from node a to node b without using arc (a,b).

STEP 3:

(a) If breakthrough occurs, use the backtracking and flow changing rules, replace $f(x, y)$ by the resultant flow in each $(x, y)\in A$, and if $f(a, b)>0$, repeat STEP 2.†

(b) Otherwise, replace $U(a, b)$ with $f(a, b)$ and if $U(a, b) \geq \max_{(x, y)\in S} U(x, y)$ terminate: arc (a, b) is a most vital link. Otherwise, replace $f^*(x, y)$ with $f(x, y)$ for all $(x, y)\in A$ and go to STEP 1.

By the use of this algorithm, a maximal flow pattern is always maintained and once non-breakthrough results for the arc (a, b) and $U(a, b) \geq U(x, y)$ for all arcs (x, y) of the candidate set S, the arc (a, b) can be declared a most vital link, i.e., non-breakthrough implies that $v(a, b)=f^*(a, b)=U(a, b)$ and, thus, $v(a, b) \geq U(x, y) \geq v(x, y)$ for $(x, y)\in S$ and by Theorem 1 the most vital link of $[N; A]$ is an element of S.

---

*The Ford and Fulkerson maximal flow algorithm [2] may be used since it identifies a minimal cut as well as a maximal flow pattern.

†An alternative to STEP 3(a) is to test if $f(x, y)<U(x, y)$, $(x, y)\in S$, and if $f(x, y) \geq U^*$, to replace $U(x, y)$ with $f(x, y)$; if $f(x, y)<U^*$ then arc (x, y) may be dropped from the set S. The computations are continued by either repeating STEP 2 if arc (a, b) has not been dropped or returning to STEP 1 if arc (a, b) has been dropped.

The other aspects of the method that must be considered are: (1) is the method finite and (2) is there a way to locate alternative optimal solutions? The finiteness of the procedure follows since there can be only a finite number of arcs in the candidate set and there is a finite number of labelings that can be made for each arc of the set. Alternative optimal solutions are readily identified as follows: after an optimal solution has been found and other candidate arcs exist, reapply the algorithm to any arc in the candidate set whose upper bound is equal to the value of the most vital link (a, b), i.e., set $U^* = U(a, b)$, delete arc (a, b) from the candidate set, and reapply the algorithm. If upon reapplication of the algorithm the most vital link has a value equal to U^*, then an alternative solution has been found. In the latter case, the algorithm is reapplied using a further reduced candidate set and the procedure is continued until all alternative optimal solutions have been found.

EXAMPLE

Consider the flow network $[N; A]$ shown in Figure 1.

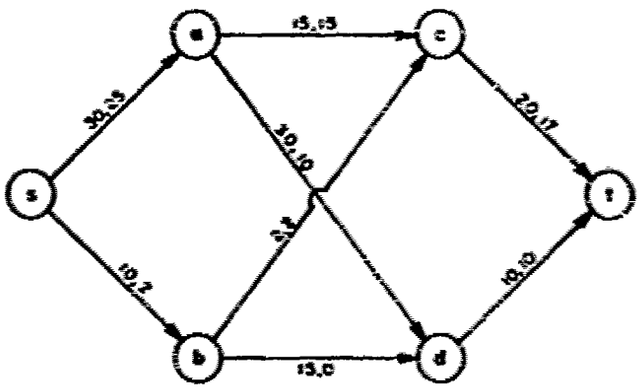


FIGURE 1. Maximal flow in sample network with source node s and sink node t.

Associated with each arc in the network is an ordered pair of numbers. The first number corresponds to the capacity of the arc and the second number corresponds to the amount of flow over the arc in some maximal flow pattern.

The improved algorithm will be employed to find all most vital links in the network $[N; A]$.

STEP 0:

(a) Figure 1 presents a maximal flow pattern F^* in the network with 27 units of flow from node s to node t. The individual arc flows are:

$$f^*(s, a) = 25, \qquad f^*(a, c) = 15.$$
$$f^*(s, b) = 2, \qquad f^*(c, d) = 10.$$
$$f^*(c, t) = 17, \qquad f^*(b, c) = 2.$$
$$f^*(d, t) = 10, \qquad f^*(b, d) = 0.$$

The corresponding minimal cut (which is unique) contains the arcs (a, c), (b, c), and (d, t). Hence,

$$U^* = \max_{(x, y) \in A^*} f^*(x, y) = f^*(a, c) = 15.$$

(b) The set of candidate arcs is:

$$S=\{(s, a), (c, t), (a, c)\}.$$

STEP 1: Start with arc (s, a) since $U(s, a)=f^*(s, a)=\max_{(x, y)\in S} U(x, y)$.

STEP 2: Labeling from node s to node a without using arc (s, a) is possible over the path containing the arcs (s, b), (b, d), and (a, d). The value of $\epsilon=8$.

STEP 3:

(a) Since breakthrough occurred, the ϵ units of flow are removed from arc (s, a) and added to each arc in the path found in STEP 2. The revised maximal flow pattern is shown in Figure 2.

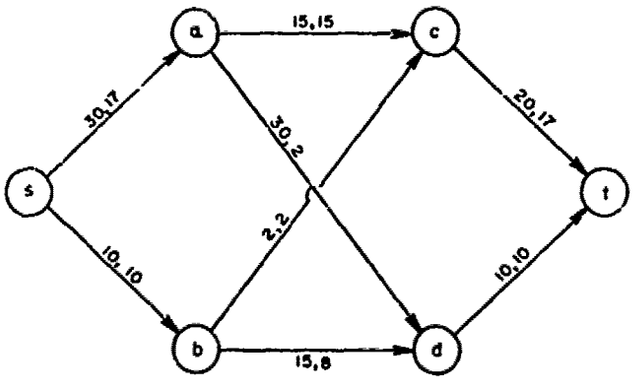


FIGURE 2. Revised maximal flow in sample network.

STEP 2: Starting with the maximal flow pattern of Figure 2. it is not possible to label from node s to node a and non-breakthrough has occurred.

STEP 3:

(b) Replace $U(s, a)$ with the flow $f(s, a)=17$. Since $U(s, a)\geq\max_{(x, y)\in S} U(x, y)=U(c, t)$, arc (s, a) is a most vital link.

Since $U(c, t)=U(c, a)=17$, arc (c, t) may also be a most vital link. In order to test this hypothesis, arc (s, a) is dropped from the set S and the algorithm is reapplied starting with the maximal flow pattern in Figure 2.

Employment of the labeling rules from node c to node t results in non-breakthrough, and arc (c, t) is an alternative most vital link.

REFERENCES

[1] E. P. Durbin, "An Interdiction Model of Highway Transportation." RM-4945-PR. The Rand Corporation. Santa Monica, California, May 1966.

[2] L. R. Ford, Jr. and D. R. Fulkerson, Flows in Networks (Princeton University Press. Princeton, N.J., 1962). pp. 17–18.

[3] R. D. Wollmer, "Some Methods for Determining the Most Vital Link in a Railway Network." RM-3321-ISA, The Rand Corporation, Santa Monica, California, April 1963.

OPTIMAL LOCATION OF A SINGLE SERVICE CENTER OF CERTAIN TYPES*

K. P. K. Nair

and

R. Chandrasekaran

Case Western Reserve University,
Cleveland, Ohio

ABSTRACT

Hakimi has considered the problem of finding an optimal location for a single service center, such as a hospital or a police station. He used a graph theoretic model to represent the region being serviced. The communities are represented by the nodes while the road network is represented by the arcs of the graph. In his work, the objective is one of minimizing the maximum of the shortest distances between the vertices and the service center. In the present work, the region being serviced is represented by a convex polygon and communities are spread over the entire region. The objective is to minimize the maximum of Euclidian distances between the service center and any point in the polygon. Two methods of solution presented are (i) a geometric method, and (ii) a quadratic programming formulation. Of these, the geometric method is simpler and more efficient. It is seen that for a class of problems, the geometric method is well suited and very efficient while the graph theoretic method, in general, will give only approximate solutions in spite of the increased efforts involved. But, for a different class of problems, the graph theoretic approach will be more appropriate while the geometric method will provide only approximate solutions though with ease. Finally, some feasible applications of importance are outlined and a few meaningful extensions are indicated.

1. INTRODUCTION

Hakimi [2] has considered a class of problems dealing with the determination of optimal location of service centers, such as hospitals and police stations. In his graph theoretic formulation of the problem, the nodes and arcs represent, respectively, the communities and road network. The optimal location obtained minimizes the maximum of the shortest distances from the service center to the vertices of the graph. A more general version of the problem has been solved by Frank [1] using the idea of a game defined on the graph. Subsequently Hakimi [3] has also considered the problem of determining the minimum number of policemen and their locations in a highway system so that the distance from any point in the highway system to the nearest policeman will not exceed a specified value.

In this paper, a convex polyhedral region is considered and communities are assumed to be spread over the entire region. Further, it is assumed that the distance between two points in region is the usual Euclidian distance. These assumptions are realistic if the transportation medium follows a straight line

*This report was prepared as part of the activities of the Department of Operations Research, School of Management, Case Western Reserve University (under Contract Number DAHC 19-68-C-0007 with Project Themis). Reproduction in whole or part is permitted for any purpose of the United States Government.

path as in the case of a helicopter. Further, this model is exact for determining the optimal location for a radio transmitting station or a radar station. It is also a very good approximation when the road network is highly developed and the simplicity of the solution method makes it attractive in these cases as well. The objective here is one of minimizing the maximum distance from the service center. The two methods of solution presented are (i) a geometric method, and (ii) a quadratic programming formulation. The appropriateness of the formulation presented in this paper and that of Hakimi [2] is discussed and the results in a specific example are compared. In certain applications the geometric method of solution is shown to be more appropriate and efficient than the graph theoretic method of solution provided by Hakimi [2]. In situations where Hakimi's solution is more appropriate, the geometric solution will be only approximate, but with considerably less effort. Some feasible applications of importance are outlined and useful extensions are indicated.

2. STATEMENT OF THE PROBLEM

Let the convex polygon, P, under consideration have a finite number of corner points numbered $1, 2, \ldots, n$. (It may be noted that a nonconvex polygon can be converted to a convex polygon by joining some of the corner points.) Since the polygon, P, is convex the largest distance from a location point, c, will be at one or more of the corner points. Define $d(i, c)$ to be the distance between c and corner point, i. Now the problem is to determine the location c so that $\max_i d(i, c)$ is minimum.

3. PROPERTIES OF THE SOLUTION

In this section some inportant properties of the solution are established so that the geometric method of solution presented in the next section will be clear.

The optimal solution to the problem can be characterized by the covering circle of the convex polygon. The covering circle of a convex polygon is defined as the smallest circle that will cover the polygon entirely. Obviously, the optimal location is at the center of the covering circle. Thus, the problem is one of determining the center c of the covering circle of the convex polygon, P.

THEOREM 1: The center of the covering circle of a convex polygon, P, is always within P.

PROOF: Consider a center c' outside P. Now find a point c in P, such that $d(c, c')$ is minimum. Obviously, the point c is unique; it could be one of the corner points or a point on one of the sides of P. Both these cases are illustrated in Figure 1. The radius of the smallest circle, with center c', that would cover P is $d(k', c')$, where k' is the corner point farthest from c'. Similarly, the smallest circle, with center c, that would cover P will have a radius of $d(k, c)$, where k is the corner point farthest from c. Since,

$$d(k, c') < d(k', c')$$

and

$$d(k, c) < d(k, c'),$$

it follows that

$$d(k, c) < d(k', c') \tag{1}$$

Inequality (1) shows that there is a smaller circle, whose center is within P, that would cover the entire polygon. This is a contradiction which completes the proof. This is illustrated in Figure 1.

THEOREM 2: The diameter of the covering circle of a convex polygon cannot be less than its largest diagonal. (The largest diagonal of a convex polygon is the straight line joining the two corner points that are farthest from each other.) The proof of this theorem is obvious.

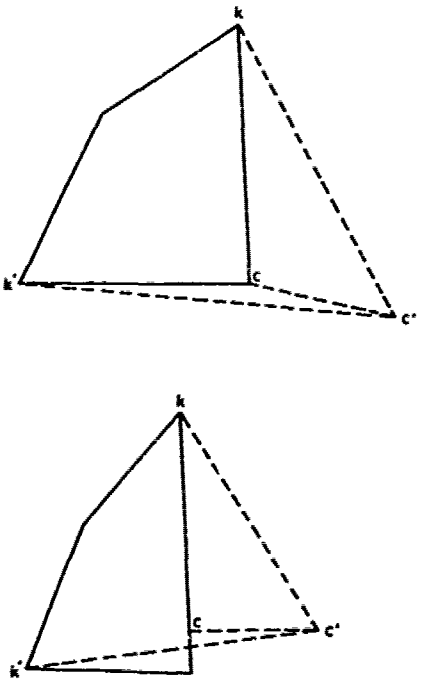


FIGURE 1. Proof of Theorem 1

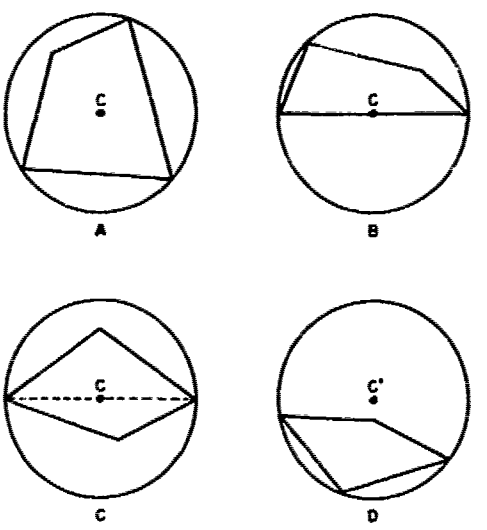


FIGURE 2. Illustration of Theorem 2
(For the respective polygons *A*, *B*, and *C* are covering circles, while *D* is not).

THEOREM 3: The covering circle of a convex polygon will pass through two or more of its corner points and all such corner points cannot be on less than half the perimeter of the circle.

The proof of this can be easily seen through contradiction. The theorem is illustrated in Figure 2.

4. METHODS OF SOLUTION

In this section, first, a geometric method of solution is presented and illustrated. Also, a quadratic program that solves the problem is provided. But no attempt is made to compare the two methods.

since, in all practical problems the geometric method will be done manually while the quadratic program will be solved on a computer. When the two methods require such different means a comparison of computation time is not meaningful. However, based on prior experience with quadratic programs, the authors strongly believe that the geometric method will be more efficient. Further, it is very simple.

Geometric Method:

Step 1: Find the farthest corner points (r, s) of P. (If there are more than one such pair take any one of them.)

Step 2: Determine the smallest of the angles subtended by the diagonal r–s at the corner points on each side of r–s. Let these angles be θ_1 and θ_2 and the corresponding corner points be v_1 and v_2, respectively. If $\theta_1 + \theta_2 \geq 180°$ go to Step 3; otherwise go to Step 4.

Step 3: The optimal location point c (i.e., the center of the covering circle) is:

(i) the midpoint of the diagonal r–s if $\theta_1 \geq 90°$ and $\theta_2 \geq 90°$

(ii) the circumcenter of the triangle rsv_1 if $\theta_1 < 90°$

(iii) the circumcenter of the triangle rsv_2 if $\theta_2 < 90°$.

(Step 3 gives the solution and hence the algorithm terminates here.)

Step 4: Find the next largest diagonal and work through the above steps.

PROOF: The proof is given for each of the three cases of Step 3 considering the nature of the solution obtained therein.

(i) The diagonal r–s is the lower bound on the diameter of the covering circle. Since $\theta_1 \geq 90°$ and $\theta_2 \geq 90°$, the circle with center at the midpoint of r–s and diameter r–s covers P, and hence it is the required circle.

(ii) The circle circumscribing the nonobtuse triangle rsv_1 is the smallest circle that can cover rsv_1 and this circle by construction covers all the corner points of P and this is therefore, the required circle.

(iii) The same argument given for (ii) above holds good.

It is of interest to note that the number of iterations in this method is finite since the number of corner points n in P is finite. In fact the solution will be either the midpoint of the largest diagonal or the circumcenter of one of the triangles which could be formed with the corner points of the polygon such that the circumcenter radius of each will not be less than $d(r, s)/2$. Therefore the number of iterations is bounded by the number of such triangles thereby making the method very efficient. Solution of an example is presented in Figures 3 and 4, showing the required iterations.

A Quadratic Programming Formulation:

As stated in Section 2 above, the problem is to determine a point c such that $\max_i d(i, c)$ is minimized. This can be formulated as a quadratic program as follows.

Let the coordinates of the corner points of P be denoted by $(y_i^1, y_i^2) i = 1, 2, \ldots, n$ and, those of c by (x^1, x^2). Also, define $d = \max_i d(i, c)$. Then,

$$d \geq d(i, c)$$

or equivalently,

$$d^2 \geq (x^1 - y_i^1)^2 + (x^2 - y_i^2)^2 \qquad i = 1, 2, \ldots, n. \tag{2}$$

Now defining a new variable $\lambda = d^2 - (x^1)^2 - (x^2)^2$, the problem reduces to the following quadratic program.

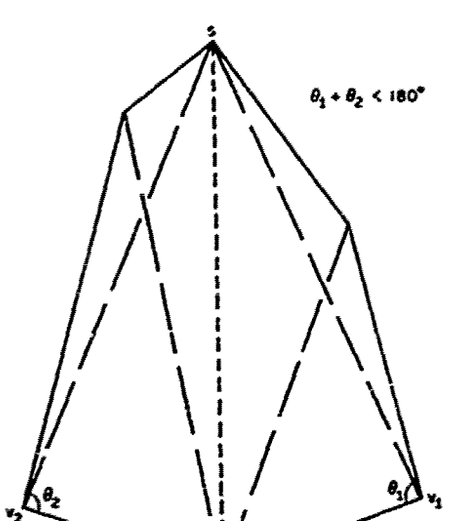


FIGURE 3. Illustration of the geometric method (first iteration)

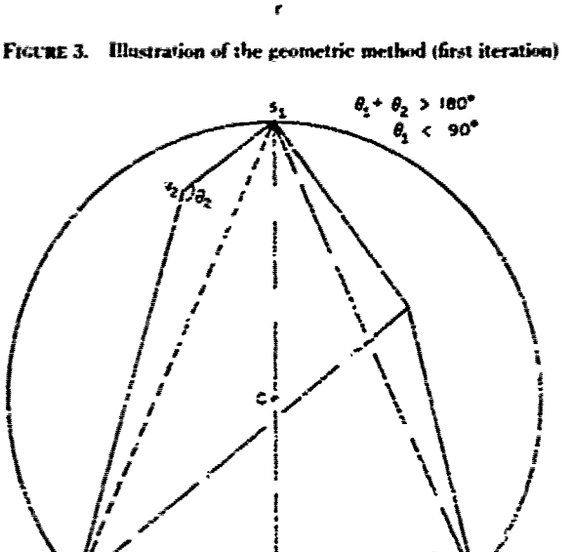


FIGURE 4. Second iteration and solution

(3) Determine λ, x^1 and x^2

minimizing $\lambda + (x^1)^2 + (x^2)^2$

subject to $2x^1y_i^1 + 2x^2y_i^2 + \lambda \geq (y_i^1)^2 + (y_i^2)^2, \quad i = 1, 2, \ldots, n.$

This quadratic program can be solved by any one of the well known methods, many of which are finite.

5. COMPARISON WITH HAKIMI'S WORK

Firstly, it should be noted that Hakimi [2] assumes that the customers are concentrated at the

various vertices of the graph which represents the region. In the present work, the customers are allowed to be spread over the entire convex polygon representing the region. Therefore, it is clear that the two problems are distinctly different; however, Hakimi's method can give a close approximation to the present problem if a very large number of vertices are introduced in the graph or a fine grid is used. But such an approach will make his method less efficient from computational point of view. For the same number of vertices, in the graph, Frank's method [1] will be a better approximation since in this formulation customers are allowed to be also on the various branches of the graph.

From the above it can be concluded that for a certain class of service centers, such as radar stations or radio transmitters, the geometric method of solution is exact and very efficient while the graph theoretic methods of Hakimi and Frank will provide only approximate solutions. Further, a good approximation to be obtained, their methods would require considerably high level of computational efforts. However, there may be certain location problems for which the graph theoretic methods are well suited and in such cases, indeed, the geometric method will provide only approximate solutions but with ease. However, it is conceivable that in some problems, the geometric method as well as the graph theoretic method may give identical or close solutions; but this is a rare event dependent upon the graphic abstraction for a given region for this to happen. Obviously one cannot base the choice of method on such a rare event.

For illustrative purpose, a triangle is considered and the solutions obtained by all the three methods are shown in Figure 5. While solving with the graph theoretic method the complete graph given by the polygon is used; no additional vertex is introduced. This approach has been suggested by one of the referees. In the example, the complete graph is therefore the very same triangle. Frank's method gives an infinite number of optimal solutions, namely, every point on the sides including the vertices, but the one which is nearest to the geometric solution is considered. The solutions obtained by the geometric method, Frank's method and Hakimi's method are denoted by G, F, and H, respectively, in Figure 5. It is seen that.

$$\max_i d(i, G) : \max_i d(i, F) : \max_i d(i, H) = 1 : 1{\cdot}5 : \sqrt{3}.$$

It should be noted that the only means of improving the solution in the graph theoretic method is by approximating the region by a fine grid, but this would involve a considerably larger amount of computational effort.

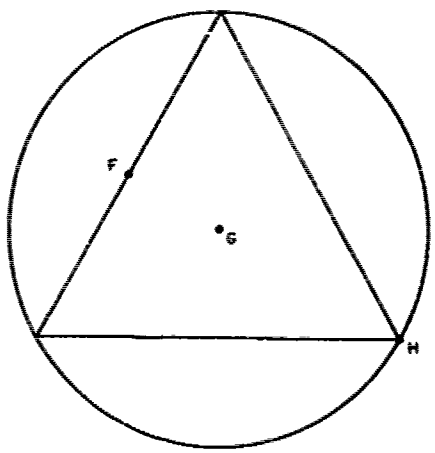


FIGURE 5. Solutions of a problem with different methods
(G—Geometric Method, F—Frank's Method, and H—Hakimi's Method)

6. FEASIBLE APPLICATIONS

(i) Location of a Radar Station or a Radio Transmitter:

The region to be scanned by a radar station could be represented by a convex polygon. Then the optimal location of the station which would minimize the maximum distance to be scanned is given by the center of the covering circle of the convex polygon. The required power of the station is dictated by the radius of the covering circle. Similar arguments will hold good in the case of a radio transmitting station also. The geometric method of solution presented is well suited for this type of problem.

(ii) Location of a Hospital for Emergency Cases:

In a combat area, base hospitals have to be located to provide emergency treatment of injured personnel. Normally the injured personnel will be brought by helicopters. Helicopters can be assumed to follow the straight path to the hospital. In this situation, the geometric method of solution is applicable.

(iii) Location of a Police Station or a Fire Station:

Police and fire station personnel normally use road vehicles. Therefore, the solution given by the geometric method will be only approximate. However, this approximation will be a good one provided the road network is well developed all over the region.

7. DISCUSSION

The translation of the optimal location problem to one of finding the covering circle of a convex polygon is of theoretical novelty and the geometric method of solution presented is simple and efficient. Solution by quadratic programming may require considerable computational effort. The characterization of the covering circle may be of interest, also, in contexts other than location problems.

The geometric method is well suited in the case of service centers, such as radar stations or radio transmitting stations. In these cases, the graph theoretic model can give only approximate solutions in spite of increased effort. Further, if the road network is well developed, this method will give a good approximation to the optimal location of service centers, such as a police station or a fire station. Although the two methods may result in the same or close locations in some problems, it is not generally true. Such a result is heavily dependent on the graphic abstraction of the region. But there may be certain problems for which only the graph theoretic model is capable of finding an accurate solution. Thus there are distinct classes of problems for which these methods may be, respectively, applied.

In this paper only a single service center is considered. Generalizations of the methods for solving the problem of locating several identical service centers will be of theoretical novelty and practical value. Further, the question optimal addition of service centers to a system already in operation is meaningful and a method for answering this question will be of immense value.

8. CONCLUSIONS

1. The optimal location of a single service center of certain types is at the center of the covering circle of the convex polygon representing the region to be served.
2. The properties of the covering circle are established and a finite geometric method of solution is presented and illustrated. Also, it is shown that the problem could be solved by a quadratic program.
3. The appropriateness of the geometric method and the graph theoretic methods available in literature is discussed.
4. The geometric method of solution has distinct applications where other methods are, in general, inappropriate and less efficient.
5. There could be some cases where the geometric method is inappropriate, but in some of these cases

it may give approximate solutions with ease. However, if a high accuracy is desired this method should not be used unless appropriate.

6. A few feasible applications are outlined besides indicating a few extensions.

REFERENCES

[1] Frank, H., "A Note on a Graph Theoretic Game of Hakimi's," Operations Research **15**, 567–570 (1967).

[2] Hakimi, S. L., "Optimum Location of Switching Centers and the Absolute Centers and Absolute Medians of a Graph," Operations Research **12**, 450–459 (1964).

[3] Hakimi, S. L., "Optimum Distribution of Switching Centers in a Communications Network and Some Related Graph Theoretic Problems," Operations Research **13**, 462–475 (1965).

SCHEDULING WITH EARLIEST START AND DUE DATE CONSTRAINTS

Paul Bratley, Michael Florian,

and

Pierre Robillard

Departement d'Informatique
Universite de Montreal

ABSTRACT

We consider the scheduling of n tasks on a single resource. Each task becomes available for processing at time a_i, must be completed by time b_i, and requires d_i time units for processing. The aim is to find a schedule that minimizes the elapsed time to complete all jobs. We present solution algorithms for this problem when job splitting is permitted and when job splitting is not permitted. Then we consider several scheduling situations which arise in practice where these models may apply.

1. INTRODUCTION

We consider the scheduling of n tasks on a single resource (machine). The tasks (jobs) become available for processing at times $a_i \geq 0$, require d_i time units for processing, and must be completed by time b_i, $i = 1, \ldots, n$. The aim is to determine whether there exists a feasible schedule that satisfies all constraints and if so to find the schedule that minimizes the total elapsed time.

Several forms of this problem have been treated earlier. Ford and Fulkerson [1, p. 65] discuss the problem when there is no job slack, i.e., $d_i = b_i - a_i$, and give a method for finding the minimum number of resources to complete all tasks. If the earliest start constraints are relaxed, i.e., if all the a_i are zero, Jackson [2] has showed that the maximum job lateness (violation of its due date) is minimized by ordering the jobs in the order of nondecreasing due dates. If the maximum lateness is zero, then all tasks may be performed using one resource and meet the due date constraints.

We shall present solution algorithms for two versions of this problem. First we allow job splitting. Under this assumption it is only required to complete d_i units of work between a_i and b_i. The smallest unit of work is the time unit considered. We show that this problem may be solved with a "labeling" type algorithm and then generalize the model to consider several resources of the same type. Then we consider the problem when no job splitting is allowed. We develop an implicit enumeration algorithm, that has very strong exclusion features, to obtain the solution of this version of the problem. Finally we shall consider several scheduling situations which arise in practice where these models may apply.

2. JOB SPLITTING PERMITTED

It is convenient to visualize this problem as follows. Consider the rectangular matrix that has a column for each job and a line for each unit of time available. There are $\max_i(b_i)$ lines and n columns. In this matrix we shall distinguish between *admissible* and *inadmissible* cells. For job i the cell (i, j) is admissible if $a_i \leq j \leq b_i$ and inadmissible otherwise. In other words the admissible cells correspond

to the time periods where the task may be performed. Time period 1 starts at time 0 and ends at time 1. An example is illustrated in Figure 1.

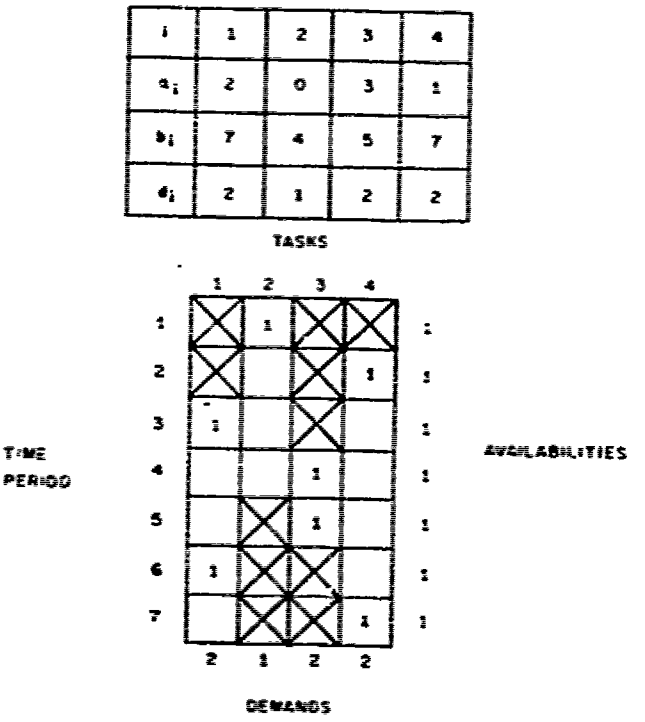


Figure 1

We associate with each row an availability of one unit (of time) and with each column a requirement of d_i. If job i is being processed at time j, a 1 placed in the admissible cell (i, j) represents this allocation. It is easy to see now that this problem is equivalent to that of finding a set of 1's placed in admissible cells such that column sums satisfy the requirements d_i and each line contains at most a single 1. In network terminology, the problem is that of finding a feasible flow in a bipartite network that has an arc for each admissible cell, sources of at most 1 unit of flow and sinks of d_i units of flow. See Figure 2.

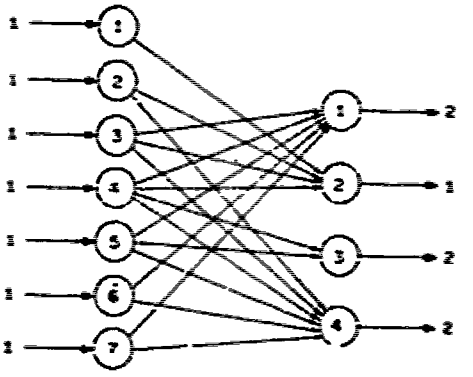


Figure 2. Network representation of Problem of Figure 1

Thus a labeling algorithm, such as that utilized to obtain a maximum flow on the admissible cells in the primal-dual algorithm for the transportation problem [1], may be employed to obtain a solution.

A slight adaptation of this algorithm is useful since each source has an availability of at most one unit and each admissible cell may contain at most a single 1.

(1) Find an initial assignment of 1's. If all the column demands are satisfied—STOP—A feasible schedule has been constructed. Otherwise continue to step 2.

(2) *Labeling Routine:* Label each line that does not contain a 1 by (−). Next select a marked line i and label all unlabeled columns j, such that (i, j) is an admissible cell with label (i). Repeat for all labeled rows. Repeat the successive row and column labeling until a column with an unsatisfied requirement is labeled (breakthrough). Go to step 3. Otherwise, if no more labels may be assigned and no breakthrough occurs *the problem is infeasible.*

(3) *Change of 1's:* In the column containing a 1 that has just been labeled, assign a 1 in the line indicated by its label. Consider next this row and remove the 1 in the column indicated by its label; repeat for all columns and lines labeled.

If all demands are satisfied—A *feasible schedule has been constructed.* Otherwise return to step 2.

The form of the solution algorithm allows the generalization of the problem to multiple resources. The availability of each line is equal to the number of resources (machines); however, each admissible cell may contain only a single 1. Furthermore, if the number of resources varies over the horizon of the scheduling problem each line may then have different availabilities.

3. JOB SPLITTING NOT PERMITTED

We shall approach this problem by considering all the possible orderings of n tasks on a single resource. There are $n!$ sequences; however, many of these are infeasible due to the violation of a due date. We propose to enumerate implicitly all the possible orderings by a branch, exclude and bound type algorithm. We describe and justify the algorithm in the following.

Branching: We enumerate the possible sequences by a tree type construction. From the initial node, or origin, of the tree we branch to n new nodes on the first level of descendant nodes. Each of these nodes represents the assignment of task i, $1 \leq i \leq n$, to be the first in the sequence. We associate with each node the completion time, t_i, of the task in this position, i.e., $t_i^1 = a_i + d_i$. Next we branch from each node on the first level to $(n-1)$ nodes on the second level. Each of these nodes represents the assignment of each of the $(n-1)$ unassigned tasks to be second in the sequence. As before, we associate the corresponding node the completion time of the task t_j^2, $t_j^2 = \max(t_i^1, a_j) + d_j$. We continue in similar fashion. In general, on level k, $1 \leq k \leq n$, there are $(n-k+1)$ new nodes generated from each node on the preceding level. It is evident that all the $n!$ orderings are enumerated in this way.

Recognizing an optimal solution: Suppose that a feasible solution has been generated. If this solution may be identified as an optimal solution then the computations may be terminated, unless one is interested in all optimal solutions. To do this we focus our attention on certain groups of tasks in a given ordering which we call blocks. A *block* is a group of jobs such that the first job starts at its earliest start time and all the following jobs to the end of the sequence are processed without any delays. The length of a block is the sum of the processing times of the tasks in the block. A block may be found by scanning the feasible solution starting from the last job in the sequence and attempting to find a group of tasks that satisfies the definition. If a block has the property that the earliest start times of all the tasks in the block are greater than or equal to the earliest start time of the first task in the block

(earliest start property), then the feasible solution found is clearly optimal. Two particular optimal solutions, if they are feasible, are the schedules of duration $a_m + d_m$, where $a_m = \max_i \{a_i\}$, or of duration $\sum_{i=1}^{n} d_i + \min_i \{a_i\}$. In the first the block consists of the job with the latest earliest start time and in the second the block consists of all the tasks to be performed.

If a block is found, but does not have the earliest start property, one may scan the schedule for the existence of a larger block that may contain the tasks of the block found. Furthermore, if the block found does not have the earliest start property, the enumeration of all other orderings for the tasks in the block may be excluded since the order of the block is the best subsequence for these tasks and one may backtrack to the level of the tree that corresponds to the first task in the block.

Exclusion: Consider the $(n-k+1)$ new nodes generated on level k of the tree construction. If the finish time t_i^k associated with at least one of these nodes exceeds its due date then all these nodes may be excluded from further consideration. The justification for this exclusion feature is the following: if any of these tasks exceeds its due date in position k of the ordering, it will certainly exceed this due date if it is scheduled later. Since all the other nodes represent orderings in which the task in question is scheduled later, they may be all omitted.

Problem Decomposition: Another means available to curtail the enumeration of solutions is to recognize that the problem splits due to the earliest start constraints. Consider level k and suppose we generate a node on that level for job i. This is equivalent to assigning job i in position k of the sequence. Let its finish time in this position be t_i^k. If t_i^k is less than or equal to the smallest earliest start time of the unscheduled jobs then the problem decomposes at level k and one need not backtrack beyond level $k+1$. The proof of this strong exclusion feature is the following. The best schedule for the remaining $(n-k)$ jobs may not be started prior to the smallest earliest start time among these jobs. Thus the order of the first k jobs cannot affect the time required to process all jobs when the stated condition occurs.

Bounding: Suppose that a feasible solution has been constructed and it is not optimal. Its value is the completion time of the last task in the sequence. Let this value be t_{fin}. We reduce then all the due dates t_i to be at most $t_{fin}-1$. This ensures that if other feasible solutions exist, only those that are better than the solution at hand are generated.

Let k be the index of the level in the tree, $i(k)$ be the index of nodes on level k and l the last level that we need to backtrack to. The detailed steps of the algorithm are as follows:

Initialize

① $k=0$, $l=0$

Generate a new level of nodes

② $k=k+1$

③ If any of the $(n-k+1)$ possible new nodes is due date infeasible, go to 7.

④ If all the nodes have been generated on this level, go to 7. Generate a new node $i(k)$ for the next unscheduled job on level k. Save $t_{i(k)}^k$. If the schedule is complete go to 6.

⑤ If the problem splits at this level, set $l=k$. Go to 2.

A feasible solution has been constructed

⑥ If a block can be found that has the earliest start property—STOP—the solution is optimal. Otherwise set $k=k'$, where k' is the level (position) of the first task in the block and return to 4.

Backtrack a level

⑦ $k = k - 1$

If $k = 1$, no further backtracking is necessary—STOP—the solution is optimal or if no solution has been constructed, the problem is infeasible.

Otherwise return to 4.

A sample problem is given in Figure 3.

i	1	2	3	4
a_i	4	1	1	0
b_i	7	5	6	4
d_i	2	1	2	2

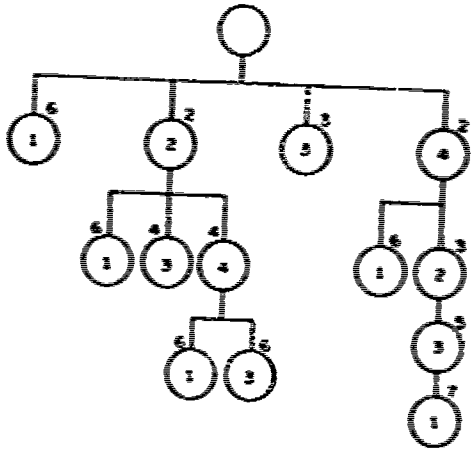


Figure 3.

We carried out tests with the algorithm developed above on a series of randomly generated problems. Test problems were generated as follows:

i) The number of jobs to be included in the problem was fixed in advance, as were three parameters a_{max}, d_{max}, and s_{max};

ii) Values of a_i were selected randomly from a uniform distribution between 1 and a_{max};

iii) Values of d_i were selected randomly from a uniform distribution between 1 and d_{max};

iv) A value, s_i say, for the slack for the i*th job* was selected randomly from a uniform distribution between 1 and s_{max}, and then b_i was calculated from $b_i = s_i + a_i + d_i$.

The time taken to solve problems of a given size was reasonably consistent, with an occasional outlier. For example, of the 50 jobs considered in the first line of Table 1, one took 0.341 sec., one 0.193 sec., one 0.120 sec., and the rest between 0.034 and 0.059 sec. This kind of distribution seems to be typical. The time taken to solve a problem apparently increases somewhat as the constraints are tightened, at least while the problem remains feasible. Infeasible problems seem to take either very

little time, or else somewhat more time than the average.

Table 1 summarises some of the experimental results obtained. The times quoted are in seconds, measured on a CDC 6400 using the optimising Fortran compiler (FTN).

TABLE 1

Number of Jobs in Problem	a_{max}	d_{max}	s_{max}	Number of Trials	Outliers: Time for Solution	Remaining Problems: Minimum Time for Solution	Remaining Problems: Maximum Time for Solution
25	200	5	50	50	0.341 0.192 [a] 0.129	0.034	0.059
25	200	10	50	25	> 150 [b] 147.975 [a] 0.860 [a] 0.010 [a]	0.025	0.204
50	500	10	100	25	5.197 1.943 1.390 1.315 0.022 [a]	0.142	0.868
100	1,000	10	200	15	> 300 [b] > 300 [b] > 300 [b] > 200 [b] 66.986	0.994	18.250

[a] Problem with no feasible solution.
[b] Problem not run to completion.

In the worst case, as can be seen from the results above, there is simply no way of avoiding the enumeration of very many possible schedules, but, in general, it seems that the exclusion, bounding, and problem-division features of the algorithm enable it to cope with problems involving a considerable number of jobs. If one bears in mind that the number of possible orderings of 25 jobs is about 1.5×10^{25}, and of 100 jobs about 9.3×10^{157}, the surprising fact is not that the algorithm occasionally requires a considerable time to find the solution of a problem, but rather that it is so often capable of finding the solution quickly.

We may tentatively conclude from the above table that the chances of being able to solve a 25 or 50 job problem in a reasonable time are good, while about one third of the 100 job problems will cause trouble.

A FORTRAN code of the algorithm may be found in the appendix.

4. APPLICATIONS

A typical situation, where a problem of this type occurs, is in the operation of a data processing installation. There is only one computer available to process the various tasks. The preparation of the

data for each task is completed at time a_i, the output must be available by time b_i, and each task requires d_i units of central processor. Typical tasks are the preparation of the payroll or the report on accounts receivable. The problem then is to select an order of the tasks that allows the completion of all tasks before their due dates. Sometimes it is possible to divide the jobs into integral units of processing time. The processing of a job may be interrupted and then restarted after the processing (partial or complete) of a different job. This situation corresponds to the assumption of section 2, that job splitting is permitted.

A similar problem may arise when it is required to allocate a single resource to activities in a CPM-PERT plan of a project. Suppose that the earliest start and latest finish times of the activities have been determined prior to the allocation of resources. Then, all jobs that are to be processed on a particular resource are singled out and one must determine whether all the jobs may be completed by using only one resource. In this problem, some jobs may be technologically ordered and there are additional constraints that some jobs must be processed in a given order relative to each other. This additional restriction may be easily accounted for in the algorithm of section 3 by adding a check of order feasibility at step 3. If any of the $(n-k+1)$ possible new nodes on level k violate the order specified then all this level of nodes may be eliminated. All the information related to the order constraints must be stored for the computations, however this can be easily accomplished.

REFERENCES

[1] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks* (Princeton University Press, Princeton, N.J., 1962).

[2] J. R. Jackson, "Scheduling a Production Line to Minimize Maximum Tardiness," Management Sciences Research Project, UCLA, Research Rept. No. 43 (Jan. 1955).

APPENDIX

This appendix presents a FORTRAN code for the resolution of the problem discussed in section 3 above, that is, for the case where job splitting is not permitted. The code has been tested on a CDC 6400, and is in fact the subroutine used to provide the timings given in Table 1. The following remarks may be made:

1. As written, the largest problem that can be solved is one with 100 jobs. For larger problems all the arrays must be redimensioned appropriately.

2. Parameters are passed through the common block. On entry to the subroutine the common arrays A, B, and D should contain appropriate values (i.e., $A(1)$ holds a_1 and so forth), and the common variable N should hold the number of jobs to be considered. On exit from the subroutine the common variable Min will hold the minimum possible completion time for the schedule, and the common array Job will hold the required order of the jobs (i.e., Job(1)=6 means that job 6 must be scheduled first, and so on). If the schedule is infeasible, Min will hold -1 on exit, and in this case the values in Job are undefined.

3. The subroutine expects nonnegative values in A, B, D, and N, but does not check that the values provided are acceptable. The values in B may be changed by the subroutine, and should therefore be saved prior to the call if they are important.

SUBROUTINE SOLVE

```
      Subroutine Solve
      Common A(100),B(100),D(100),N,Min,Job(100)
      Dimension In(100),Nowtab(100),Jobtab(100)
      Integer A,B,D,Tot                          Return best value in MIN, best order in JOB
      Logical In,Newfrz
      Min = Last = -1
      Lfrz = Tot = 0                             Initialize
      Now = A(1)
      Do 1 I = 1,N
      If (D(I).GT.B(I)-A(I)) Return
      If (A(I).LT.Now) Now = A(I)
      If (B(I).GT.Last) Last = B(I)
      In(I) = .False.
    1 Tot = Tot + D(I)                           Check feasibility of individual constraints
      If (Now+Tot.GT.Last) Return
      L = 1
      Now = Now - 1
      Go To 2
    3 Newfrz = .True.                            3: Try a new level
      Do 4 J = 1,N
      If (In(J)) Go To 4
      If (Now + D(J).GT.B(J)) Go To 16           If any of the remaining jobs is infeasible, back up.
      If (Now.GT.A(J)) Newfrz = .False.          Check for the possibility of splitting the problem.
    4 Continue
      If (Newfrz) Lfrz = L
      L = L + 1
      Tot = Tot - D(I)                           New level search begins
    2 I = 1
   11 If (In(I)) Go To 5
      Newnow = Max0(Now,A(I))
      If (Newnow + Tot .GT. Last) Go To 5        Can we do job i next?
      Jobtab(L) = I
      If (L.Eq.N) Go To 6                        If so, add it to the schedule
      Nowtab(L) = Now                            If the schedule is complete, go to 6.
      Now = Newnow + D(I)
      In(I) = .True.                             Otherwise mark the newly added job, save the
      Go To 3                                      time, and go to 3 to try the next level
    6 Min = Last = Newnow + D(I)
      Do 10 J = 1,N                              6: New solution
      If (B(J) .GE. Min) B(J) = Min - 1
   10 Job(J) = Jobtab(J)
      If (Newnow.Eq.A(I)) Return                 Alter constraints and save new solution
      Now = A(I)
    7 L = L - 1
      I = Jobtab(L)
      If (Nowtab(L)-A(I)) 12,13,14

   12 If (Now.GE.A(I)) Return                    Work back along the schedule until the front of a
      Go to 9                                      block
   13 If (Now.GE.A(I)) Return                    If this last block cannot be improved, the schedule
      Go to 15                                     is optimal
   14 If (Now.GT.A(I)) Now = A(I)
   15 In(I) = False                              Otherwise try to improve the last block by finding
      Tot = Tot + D(I)                             an alternative first element
      Go To 7
    9 L = L - 1                                  9: Back up a level
      IF (L.Eq.Lfrz) Return                        If the problem splits at this level, there is no need
      I = Jobtab(L)                                to try different permutations of earlier jobs.
    8 Tot = Tot - D(I)
```

```
16  Now = Nowtab(L)
    In(I) = .False.
 5  I = I + 1
    If (I.Le.N) Go To 11
    Go To 9
    End
```

Try next alternative at the current level if there is one, otherwise back up a level

LARGE DEVIATION PROBABILITIES FOR ORDER STATISTICS

Stephen A. Book

California State College, Dominguez Hills
Dominguez Hills, California

ABSTRACT

Asymptotic representations are found for the large deviation probabilities that the $n\alpha$-th order statistic exceeds δ, where $\delta>\alpha$. The probabilities are first expressed in terms of the empirical distribution function, and then the 1960 theorem of Bahadur and Ranga Rao is applied. The result is then shown to be more precise than a logarithmic statement in a 1969 paper of Sievers dealing with the asymptotic relative efficiency of the sample median test.

1. INTRODUCTION

If $U_1, U_2, \ldots$ is a sequence of independent random variables, uniformly distributed on the interval $[0, 1]$, with order statistics $U_{([n\alpha])}$, $0<\alpha\leq 1$, for each positive integer n, we would like to find an asymptotic representation of the large deviation probabilities $P(U_{([n\alpha])}>\delta)$ for $\delta>\alpha$. ($[y]$ is the greatest integer $\leq y$). Our result extends to random variables $X_1, X_2, \ldots$ having arbitrary distribution F in view of the fact that $P(X_{([n\alpha])}>\delta)=P(F(X_{([n\alpha])})\geq F(\delta))=P(U_{([n\alpha])}>F(\delta))$. To determine the asymptotic formula, we write the large deviation probability for the order statistic in terms of a large deviation probability for the empirical distribution function (e.d.f.), and we then apply the result of Bahadur and Ranga Rao [1]. It will then remain only to compute the values of the components of the Bahadur-Ranga Rao formula.

2. THE THEOREM

From elementary results on order statistics and the representation of the e.d.f. as a binomial random variable, we obtain the following result:

LEMMA 1: If $0<\epsilon<\delta\leq 1$, and F_n is the e.d.f. of the random variables $U_1, U_2, \ldots$, then

$$P(U_{([n\alpha])}>\delta)=P(F_n(1-\delta)-(1-\delta)>\epsilon). \tag{1}$$

PROOF: We set $m=[n(1-\delta+\epsilon)]$ to simplify the notation. The result in [3] and changes of variable show that

$$P(U_{([n\alpha])}>\delta)=\sum_{j=m+1}^{n}\binom{n}{j}(1-\delta)^j\delta^{n-j}. \tag{2}$$

On the other hand, denoting the indicator function of a set A by I_A, we have that

$$\begin{aligned} P(F_n(1-\delta)-(1-\delta)>\epsilon) &= P(F_n(1-\delta)>1-\delta+\epsilon) \\ &= P\left(\sum_{k=1}^{n} I_{(U_k\leq 1-\delta)}>n(1-\delta+\epsilon)\right) \\ &= \sum_{j=m+1}^{n}\binom{n}{j}(1-\delta)^j\delta^{n-j}. \end{aligned} \tag{3}$$

Comparing (2) and (3), we obtain (1).

Now we are ready to apply the theorem of Bahadur and Ranga Rao. In [1], the two authors demonstrated, for $x \geq 0$ and $\epsilon > 0$, the existence of a positive number $\rho(x, \epsilon) < 1$ and a bounded sequence of numbers $b_n(x, \epsilon)$, such that

$$P(F_n(x) - x > \epsilon) \sim (2\pi n)^{-1/2} \rho^n(x, \epsilon) b_n(x, \epsilon)$$

where the symbol "~" means that the ratio of the two sides tends to 1 as $n \to \infty$. Among other things, the authors also gave methods for explicitly computing $\rho(x, \epsilon)$ and $b_n(x, \epsilon)$. Combining Lemma 1 with the theorem of Bahadur and Ranga Rao, we get:

COROLLARY 1: If $0 < \alpha < \delta \leq 1$, then

$$P(l'_{n-[n(1-\alpha)]} > \delta) \sim (2\pi n)^{-1/2} \rho^n(1-\delta, \delta-\alpha) b_n(1-\delta, \delta-\alpha) \tag{4}$$

where $n - [n(1-\alpha)] = n\alpha$ if $n\alpha$ is an integer, and $n - [n(1-\alpha)] = [n\alpha] + 1$ if $n\alpha$ is not an integer.

It remains to compute explicitly the numbers $\rho(1-\delta, \delta-\alpha)$ and $b_n(1-\delta, \delta-\alpha)$, which we shall abbreviate as ρ and b_n from now on.

LEMMA 2: $\rho = (\delta/\alpha)^\alpha ((1-\delta)/(1-\alpha))^{1-\alpha}$.

PROOF: $P(F_n(1-\delta) - (1-\delta) > \delta - \alpha) = P(n^{-1}\Sigma_{k=1}^n Y_k > \delta - \alpha)$, where $Y_k = I_{[U_k < 1-\delta]} - (1-\delta)$. Bahadur and Ranga Rao showed on p. 1015 of [1] that if $\varphi(t)$ is the moment-generating function of Y_1, then

$$\rho = \inf_{t>0} \exp(-(\delta-\alpha)t)\, \varphi(t).$$

Here $\varphi(t) = \exp(-(1-\delta)t)(e^t(1-\delta) + \delta)$ so that $\rho = \inf_{t>0} \exp(-(1-\alpha)t)(e^t(1-\delta) + \delta) = (\delta/\alpha)^\alpha ((1-\delta)/(1-\alpha))^{1-\alpha}$, the minimum being attained at $t = \log(\delta(1-\alpha)/\alpha(1-\delta))$.

NOTE: Carsrud, in [2], has determined the value of ρ by an entirely different method, due to Huber, involving the minimization of a certain convex function.

The method of finding b_n in [1] is somewhat more complicated, and results in the next lemma:

LEMMA 3: $b_n = ((1-\alpha)/\alpha)^{1/2} (\delta/(\delta-\alpha))\, (\alpha(1-\delta)/\delta(1-\alpha))^{n\alpha - [n\alpha]}$.

PROOF: According to the argument of Ref. [1] (pp. 1022–4) culminating in Equation (46) (on p. 1024), b_n is given by the formula

$$b_n = \frac{d \exp(-\tau d(n\alpha d^{-1} - [n\alpha d^{-1}]))}{\sigma(1 - e^{-\tau d})}, \tag{5}$$

where d is the maximum span of the lattice variable Y_1 and $\sigma^2 = (\varphi''(\tau)/\varphi(\tau)) - (\delta-\alpha)^2$ is the variance of the "associated" distribution introduced in Ref. [1] (bottom of p. 1016). Clearly $d = 1$, and we calculate from $\varphi(t)$ and $\tau = \log(\delta(1-\alpha)/\alpha(1-\delta))$ that $\sigma^2 = \alpha(1-\alpha)$. Inserting these values for d, τ, and σ into (5), we obtain the desired expression for b_n.

Having found formulas for the quantities that appear in Corollary 1, we are now ready to write down the asymptotic representation of the large-deviation probabilities for order statistics:

THEOREM: If $0 < \alpha < \delta \leq 1$, then

$$P(l_{n-[n(1-\alpha)]} > \delta) \sim (2\pi n)^{-1/2} \left(\frac{\delta}{\alpha}\right)^{n\alpha} \left(\frac{1-\delta}{1-\alpha}\right)^{n(1-\alpha)} \left(\frac{1-\alpha}{\alpha}\right)^{1/2} \left(\frac{\delta}{\delta-\alpha}\right) \left(\frac{\alpha(1-\delta)}{\delta(1-\alpha)}\right)^{n\alpha - [n\alpha]}$$

PROOF: This is an immediate consequence of Corollary 1 and Lemmas 2 and 3.

3. APPLICATION TO THE MEDIAN TEST

If F is a distribution function with a continuous symmetric density, such that $F'(0) > 0$, and $X_1, X_2, \ldots$ is a sequence of observations from $F(x-\theta)$, then the median test can be used to test the hypothesis $H_0: \theta = \theta_0$ against the alternative $H_1: \theta > \theta_0$. After transforming to the uniform distribution, we reject the hypothesis if the median $M_n = U_{([n/2]+1:n)} > \delta$ for some $\delta > \theta_0$. The probability $P(M_n > \delta)$ under the hypothesis is the significance level of the test, and as $n \to \infty$, the tests based on the statistics M_n improve in the sense that the significance levels tend to 0.

The Bahadur "exact slope" measures the rate of convergence to 0 of the significance levels of the tests. The exact slope here is $2e(\delta)$, where $e(\delta) = -\lim_{n\to\infty} n^{-1} \log P(M_n > \delta)$. Two tests of the same hypothesis may be compared by computing the ratio of their exact slopes, which is called the Bahadur asymptotic relative efficiency.

In a discussion of the asymptotic relative efficiency of the median test in Ref. [4] (p. 1914), Sievers shows that

$$\lim_{n\to\infty} n^{-1} \log P(M_n > \delta) = \tfrac{1}{2} \log (4\delta(1-\delta)). \tag{6}$$

The theorem obtained in the previous section yields a more precise measure of the rate of convergence to 0 of the significance levels of the tests.

As a particular case of the theorem, consider the large deviation probability for the median. Here $\alpha = 1/2$ and we assume, as is customary in this situation, that n takes only odd values. Then $n\alpha = n/2$ is not an integer, so our method gives the large deviation probability for $U_{([n\alpha]+1:n)} = U_{([n/2]+1:n)} = M_n$, which is the median. We have, setting $\alpha = 1/2$ in the theorem:

COROLLARY 2: For $\delta > 1/2$ and n an odd integer,

$$P(M_n > \delta) \sim (2\pi n)^{-1/2} (4\delta(1-\delta))^{n/2} (\delta(1-\delta))^{1/2} (\delta - 1/2)^{-1}. \tag{7}$$

The assertion of this corollary provides the desired asymptotic measure with somewhat more precision than does (6). Also, Sievers' result follows from (7) upon taking logarithms, dividing by n, and letting $n \to \infty$.

4. ACKNOWLEDGMENT

This paper represents part of the author's doctoral dissertation written under Prof. Donald R. Truax at the University of Oregon, Eugene.

The author would like to thank Professor Truax for suggesting the problem and for subsequent discussions on the topic.

REFERENCES

[1] Bahadur, R. R. and R. Ranga Rao, "On Deviations of the Sample Mean," Annals of Mathematical Statistics **31**, 1015–1027 (1960).

[2] Carsrud, W., Ph.D. Thesis, University of Oregon (1971).

[3] Noether, G. E., "An Identity in Probability," American Mathematical Monthly **71**, 903–904 (1964).

[4] Sievers, G. L., "On the Probability of Large Deviations and Exact Slopes," Annals of Mathematical Statistics **40**, 1908–1921 (1969).

A BIVARIATE NORMAL THEORY MAXIMUM-LIKELIHOOD TECHNIQUE WHEN CERTAIN VARIANCES ARE KNOWN

Mitchell O. Locks

College of Business Administration
Oklahoma State University

ABSTRACT

A maximum-likelihood technique is described for estimating the bivariate normal distribution of the estimates of two or more related values when data are obtained from several different sources, each having known variance. The problem is comparable, in the bivariate sense, to estimating the mean of a normal population with known variance. The results tend to be dominated by those sources of data associated with the smallest variances.

INTRODUCTION

For heavily instrumented systems, such as guided missiles and space vehicles, the values of some of the parameters of the system can frequently be obtained in several different ways because of both the volume and the variety of measurements. A given value, for example, can be obtained either from one or more direct readings, or else from reconstructions of related data. It is also frequently assumed that the error tolerances, such as the normal-theory 3-sigma errors, associated with both the measurements and the reconstructive techniques are known and can be specified in advance.

A case in point is the mass of a staged launch vehicle. The mass at engine ignition and also at cutoff can be estimated using both sensor and probe data, each with a specified error level. In addition, flowmeter data provides the difference between ignition mass and cutoff mass, and simulated reconstruction of the trajectory combined with engine thrust data provide a ratio of ignition to cutoff mass. This example, which motivated the study, is covered in more detail at the end of the paper. The technique described herein is essentially the same as one that has been used for this type of mass estimation for certain Apollo flights.

The purpose of this report is to describe a technique for estimating the bivariate normal distribution of two correlated but unknown values, using combined data from different kinds of sources, each having known error. Some of these sources are independent measurements of either value; others are reconstructions of a linear functional relationship between the two values, which are estimated by using independent data sources. The separate measurements of the predictor, the value which is measured with smallest error, are designated x and the true value is μ; the corresponding designations for the predictor are y and ν, respectively. The functional relationships are assumed to be regressions "$\nu = m\mu + b$." These can all be different regressions because they are based upon different kinds of supporting physical theory and data.

The problem is comparable, in the bivariate sense, to estimating the value of a normal-theory measurement, using data from different sources each having known error.

A maximum-likelihood (least squares) normal theory method is described for obtaining the estimates, $\hat{\mu}$ and $\hat{\nu}$, and their joint bivariate normal distribution. The reciprocals of the variances which

are proportional to the squares of the tolerance levels for the measurement systems are used as weights. The assumption is made that all systems are unbiased, or at least corrected for bias. From these data one also obtains a confidence ellipse for (μ, ν) at any specified confidence level, with the center of the ellipse being $(\hat{\mu}, \hat{\nu})$.

The shape and the orientation of the likelihood ellipsoid depend only upon the variances and the slopes, m_j, of the regressions, and not upon the measurements or the positioning constants, b_j. The bivariate normal distribution has the same shape, orientation, and position as that of the likelihood function, but differs from it only in height. As a result the estimates are influenced most by the data with the smallest errors as might be expected. Likewise, if the error specified for any of the regressions is small, it can tend to dominate the results.

DESCRIPTION

Let $x_1, \ldots, x_k$ be the k unbiased independent measurements of μ, with corresponding variances $\sigma_1^2, \ldots, \sigma_k^2$ (e.g., σ_j is one-third the maximum absolute error in the measurement of x_j); and let $y_{k+1}, \ldots, y_n$ be $n-k$ unbiased independent measurements of ν with variances $\sigma_{k+1}^2, \ldots, \sigma_n^2$. Also for $j = n+1, \ldots, p$ let b_j be normally distributed with mean $\nu - m_j\mu$ and variance σ_j^2.

The corresponding normal-theory probability functions are respectively:

$$f(x_j) = (2\pi)^{-1/2}(\sigma_j)^{-1} \exp -\tfrac{1}{2}\{(x_j - \mu)/\sigma_j\}^2, \qquad j = 1, \ldots, k$$

$$f(y_j) = (2\pi)^{-1/2}(\sigma_j)^{-1} \exp -\tfrac{1}{2}\{(y_j - \nu)/\sigma_j\}^2, \qquad j = k+1, \ldots, n$$

$$f(b_j|\mu, \nu) = (2\pi)^{-1/2}(\sigma_j)^{-1} \exp -\tfrac{1}{2}\{(b_j - \nu + m_j\mu)/\sigma_j\}^2, \qquad j = n+1, \ldots, p.$$

The joint probability or likelihood of all measurements and calculations is

$$f = \prod_{j=1}^{k} f(x_j) \prod_{j=k+1}^{n} f(y_j) \prod_{j=n+1}^{p} f(b_j|\mu, \nu)$$

$$= (2\pi)^{-p/2}\left\{\prod_{j=1}^{p} \sigma_j\right\}^{-1} \exp -\tfrac{1}{2}\, G(\mu, \nu). \tag{1}$$

$$G(\mu, \nu) = A\mu^2 + B\mu + C\nu^2 + D\mu + E\nu + F; \tag{2}$$

$$A = \sum_{j=1}^{k} 1/\sigma_j^2 + \sum_{j=n+1}^{p} m_j^2/\sigma_j^2$$

$$B = -2 \sum_{j=n+1}^{p} m_j/\sigma_j^2$$

$$C = \sum_{j=k+1}^{p} 1/\sigma_j^2$$

$$D = -2 \sum_{j=1}^{k} x_j/\sigma_j^2 + 2 \sum_{j=n+1}^{p} m_j b_j/\sigma_j^2$$

$$E = -2 \sum_{j=k+1}^{n} y_j/\sigma_j^2 - 2 \sum_{j=n+1}^{p} b_j/\sigma_j^2$$

$$F = \sum_{j=1}^{k} x_j^2/\sigma_j^2 + \sum_{j=k+1}^{n} y_j^2/\sigma_j^2 + \sum_{j=n+1}^{p} b_j^2/\sigma_j^2.$$

It can be shown that G is an ellipsoid since $4AC - B^2 > 0$; therefore $\mathcal{L}$ is also an ellipsoid.

The maximum-likelihood estimate (m.l.e.), $(\hat{u}, \hat{v})$, of the point (μ, ν) is the apex of $\mathcal{L}$ obtained by partially differentiating it, first with respect to μ and then for ν, to obtain two simultaneous equations in two unknowns

$$\frac{\partial \mathcal{L}}{\partial \mu} = -(2A\mu + B\nu + D)\mathcal{L}/2$$

$$\frac{\partial \mathcal{L}}{\partial \nu} = -(B\mu + 2C\nu + E)\mathcal{L}/2.$$

Setting both equations to zero and solving, the m.l.e. is obtained

$$\hat{\mu} = \frac{2CD - BE}{B^2 - 4AC}$$

$$\hat{\nu} = \frac{2AE - BD}{B^2 - 4AC}. \tag{3}$$

This estimate is unbiased since

$$\begin{aligned} E(-D) &= (2A\mu + B\nu) \\ &= 2A\mu + B\nu; \end{aligned}$$

likewise

$$\begin{aligned} E(-E) &= (B\mu + 2C\nu) \\ &= B\mu + 2C\nu. \end{aligned}$$

Therefore,

$$E(\hat{\mu}) = \mu; \qquad (\hat{\nu}) = \nu.$$

In the sequel the point $(\hat{\mu}, \hat{\nu})$ which is the summit of the ellipsoid is also taken to be the summit of the bivariate normal distribution which has the same shape and orientation as $\mathcal{L}$, but a different height.

The variances and covariance of these estimates are obtained as follows.* First we note that A, B, and C are independent of the data since they depend only upon known values of m_j and σ_j^2, but D and E are random functions. From (2) we have

$$\operatorname{Var}(D) = 4\left(\sum_{j=1}^{k} 1/\sigma_j^2 + \sum_{j=k+1}^{p} m_j^2/\sigma_j^2\right) = 4A$$

$$\operatorname{Var}(E) = 4 \sum_{j=k+1}^{p} 1/\sigma_j^2 = 4C$$

$$\operatorname{Cov}(D, E) = -4 \sum_{j=k+1}^{p} m_j/\sigma_j^2) = 2B.$$

Hence from (3)

$$\operatorname{Var}(\hat{\mu}) = \frac{4C^2 \operatorname{Var}(D) + B^2 \operatorname{Var}(E) - 4BC \operatorname{Cov}(D, E)}{(B^2 - 4AC)^2}$$

$$= \frac{4C}{4AC - B^2}$$

---

*I should like to thank my colleagues at Oklahoma State University, Dr. J. Leroy Folks, Dr. David Wallace of the University of Chicago, and a referee of the NLRQ for their assistance in helping me to simplify the presentation in this section, and for pointing out certain other aspects of the paper that needed more clarification.

$$\text{Var}(\hat{\nu}) = \frac{4A^2\,\text{Var}(F) + B^2\,\text{Var}(D) - 4AB\,\text{Cov}(D, E)}{(B^2 - 4AC)^2}$$

$$= \frac{+A}{4AC - B^2}$$

$$\text{Cov}(\hat{\mu}, \hat{\nu}) = \frac{(B^2 + 4AC)\,\text{Cov}(D, E) - 2BC\,\text{Var}(D) - 2AB\,\text{Var}(E)}{(B^2 - 4AC)^2}$$

$$= -\frac{2B}{4AC - B^2}.$$

It is particularly interesting to note that the variances and the covariance depend only upon the prespecified σ's and m's, but that the estimates $\hat{\mu}$ and $\hat{\nu}$ depend upon the observable data as well. Following Hald [1] it can easily be seen that the angle formed by this ellipsoid is

$$\phi = \tfrac{1}{2} \arctan\left(\frac{2\,\text{Cov}(\hat{\mu}, \hat{\nu})}{\text{Var}(\hat{\mu}) - \text{Var}(\hat{\nu})}\right), \text{Var}(\hat{\mu}) \neq \text{Var}(\hat{\nu}).$$

EXAMPLE

At the time of ignition of the second stage (S–II) of a Saturn–5 launch vehicle for an Apollo spacecraft, over two-thirds of the mass of the entire vehicle, including the second and third stages and the spacecraft payload, consisted of second-stage propellants alone. Two different measuring systems were used to estimate the amount of second-stage liquid propellant: (*i*) level sensors in the sides of the tanks, and (*ii*) capacitance probes. There is a substantial amount of variability in these measurements, more, for example, than in all of the other relevant mass data (dry weight of the vehicle plus the propellants and other consumables in the third stage and payload). Thus, the errors or tolerances associated with the measurement of the mass of the vehicle at the time of second-stage ignition are for all practical purposes the same as those associated with the measurement of the second-stage liquid level.

At the time of second-stage cutoff the amount of residual propellants could have been measured by the sensor and probe systems, as at ignition, but with considerably less variability in the readings because the tanks were nearly empty. Therefore in analyzing the relationship between cutoff mass and ignition mass, the former is used as the "predictor" and the latter as the "predictand."

Besides the sensor and probe readings, other kinds of data are available with respect to the relationship between the ignition and cutoff masses. These are:

(i) Flowmeter data (i.e., the difference between ignition mass and cutoff mass is the integral of the flows).

(ii) Trajectory reconstruction. Radar tracking data of position, acceleration and velocity, and engine thrust and specific impulse data are used, by a simulation technique, to obtain the most probable ratio of the ignition mass to the cutoff mass.

Let x_1 denote the cutoff level-sensor reading and x_2 the cutoff probe reading. Both of these are assumed to be independent unbiased estimates of the cutoff mass μ. Similarly, the ignition point-sensor reading is y_1 and the probe y_2; these are assumed to be independent unbiased estimates of the ignition mass ν. Corresponding to the maximum (3σ) errors of these values are the standard deviations σ_{x_1}, σ_{x_2}, σ_{y_1}, and σ_{y_2}.

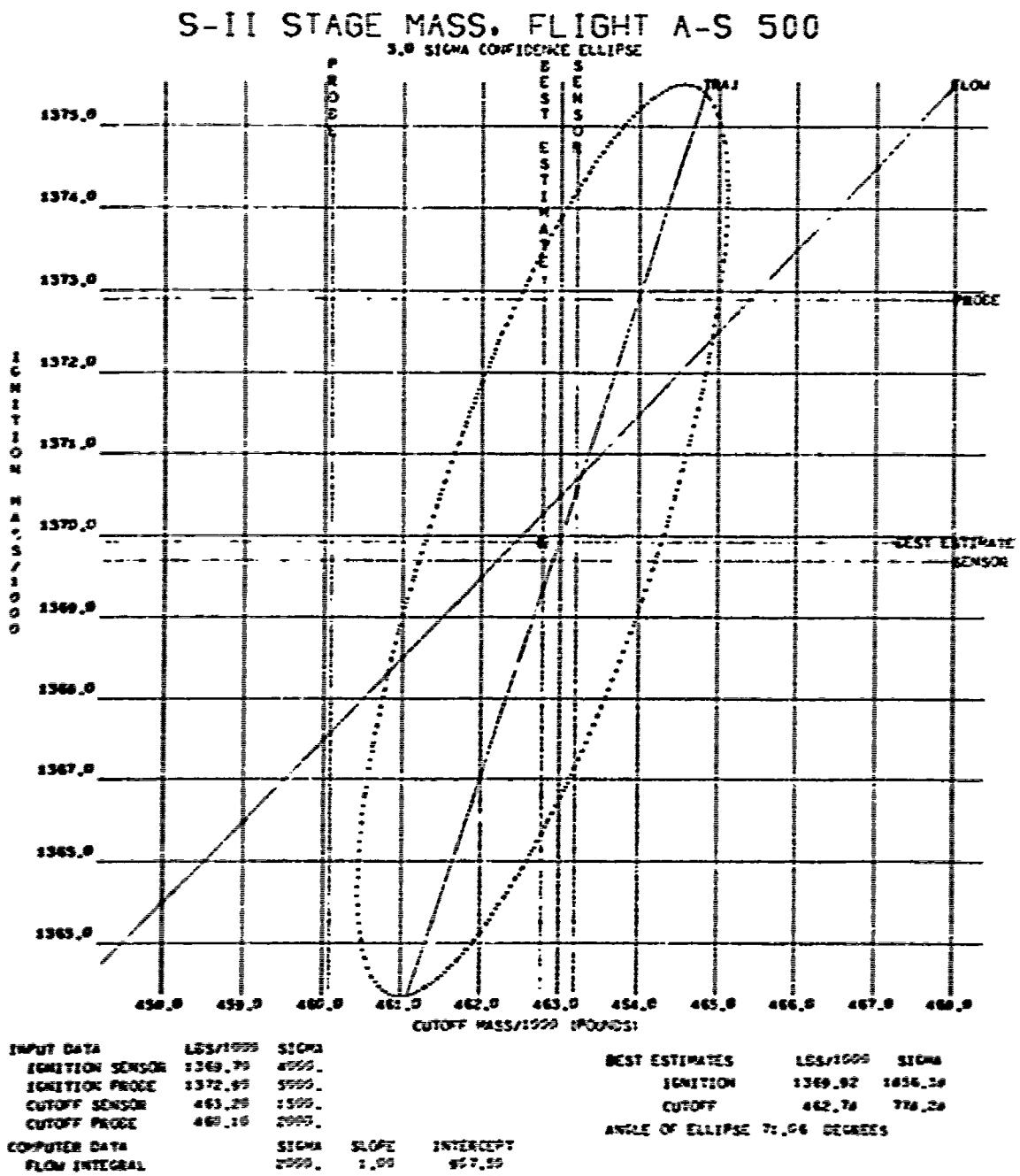


FIGURE 1. CRT Display, S-II stage mass estimates and confidence ellipse.

Let the constant b_2 denote the total (integral) of all propellant flows recorded by the flowmeters. Then it is easy to see that based upon these data we have the function

$$\nu = \mu + b_2,$$

with standard deviation σ_{y_2}. Note that the slope, m_2, is one in this case. Similarly, corresponding to the trajectory we have another function

$$\nu = m_4\mu + b_4,$$

with corresponding errors σ_{y_4}. Since reconstruction gives the ratio of the two masses, b_4 is zero.*

The hypothetical data are as follows:

	Readings	*Pounds*	*Sigma*
x_1:	Ignition sensor	1,369,700	4000
x_2:	Ignition probe	1,372,900	5000
y_1:	Cutoff sensor	463,200	1500
y_2:	Cutoff probe	460,100	2000

Other data	*Sigma*	*Slope*	*Intercept*
Flowmeter	2000	1.00	907.50
Reconstruction	1700	2.95	0.90

Based upon these data the following results were obtained:

$$\hat{\mu} = 1570; \quad \sigma_{\hat{\mu}}^2 = 1856$$
$$\hat{\nu} = 4463; \quad \sigma_{\hat{\nu}}^2 = 780$$

The angle of the confidence ellipse connecting μ and ν is 71.06 degrees. Figure 1 shows the CRT output of a 3σ confidence ellipse produced by the computer program for these data. It should be noted that because of the relatively small value for σ_{y_4}, the reconstruction data tend to dominate the results of all calculations.

REFERENCES

[1] Hald, A., *Statistical Theory With Engineering Applications* (John Wiley and Sons, New York, N.Y., 1953).

*Strictly speaking, the use of the trajectory reconstruction technique for the purpose is an approximation. Since b_4 is always fixed at zero, the random variable should be m_4 rather than $\hat{b}_4$. To introduce this correction would require a more complex model.

ON THE USE OF STANDARD TABLES TO OBTAIN DODGE-ROMIG LTPD SAMPLING INSPECTION PLANS

William C. Guenther

University of Wyoming
and
University of Oslo

ABSTRACT

Procedures are described which yield single and double sample Dodge-Romig [1] lot tolerance percent defective (LTPD) rectifying inspection plans. For the determination of such plans only a desk calculator and standard tables of the discrete probability distributions are required. Some advantages gained by using these procedures rather than the Dodge-Romig table include: (a) The Consumer's Risk is not limited to 0.10, (b) More choices of LTPD are available, (c) Smaller average total inspection is achieved by using a plan designed for specific "process average" and lot size rather than a compromise plan designed to cover intervals on these two parameters.

1. INTRODUCTION

A product that is mass produced is assembled at random into lots of size N. From each lot items are sampled at random and the number of defectives is observed. If the lot is accepted all defective items found when sampling are replaced by nondefectives. If the lot is rejected all N items are examined and all defective items in the lot are replaced. The procedure just described is the special case of rectifying inspection which we are about to consider. Our goal is to determine reasonable sampling plans for the type of situation just described.

Let us assume that the process produces a defective with probability p. Each inspected lot will contain an unknown number of defectives, say k. Let Y be the number of defectives in a random sample of size n drawn from a lot. It is well known that the probability function of Y given k is the hypergeometric

$$p(N, n, k, y) = \frac{\binom{k}{y}\binom{N-k}{n-y}}{\binom{N}{n}}, \qquad a \leq y \leq b, \tag{1.1}$$

where $a = \max [0, n - (N - k)]$, $b = \min [k, n]$, and the unconditional probability function of Y is the binomial

$$b(y; n, p) = \binom{n}{p} p^y (1-p)^{n-y}, \qquad y = 0, 1, 2, \ldots, n. \tag{1.2}$$

For both single and double sampling we will minimize the average total inspection if $p = \beta_1$, the "process average," subject to the condition that the operating characteristic (OC) of the sampling plan be no more the β_1 if the lot contains $k_1 = Np_1$ defectives. (In the language of Dodge-Romig [1] $p_1 = p_t = LTPD$, $\beta_1 =$ The Consumer's Risk.)

Binomial cumulative sums will be denoted by

$$E(r; n, p) = \sum_{y=r}^{n} b(y; n, p). \tag{1.3}$$

For our purposes we find that three tables are useful. The Ordnance Corps [7] table gives (1.3) to seven decimal places for $n = 1(1)150$, $p = 0.01(0.01)0.50$. The Harvard [4] table gives (1.3) to five decimal places for $n = 1(1)50(2)100(10)200(20)500(50)1000$, $p = 0.01(0.01)0.50$ (plus a few rational fractions). The Weintraub [8] table gives the same sum to 10 decimal places for $n = 1(1)100$, $p = 0.0001$, $0.0001(0.0001)0.001(0.001)0.10$.

In the hypergeometric case we will use the tables of Lieberman and Owen [5] which gives both (1.1) and

$$P(N, n, k, r) = \sum_{y=a}^{r} p(N, n, k, y), \tag{1.4}$$

to six decimal places for $N = 1(1)50(10)100$. In addition two approximations to (1.4) will be used. These are

$$P(N, n, k, r) \cong 1 - E\left(r+1; n, \frac{k}{N}\right), \tag{1.5}$$

if $n/N \leqq 0.10$, $k \geqq n$, and

$$P(N, n, k, r) \cong 1 - E\left(r+1; k, \frac{n}{N}\right), \tag{1.6}$$

if $k/N \leqq 0.10$, $k < n$. Even when neither condition, $n/N \leqq 0.10$ and $k/N \leqq 0.10$, is satisfied the approximation is usually surprisingly good if we use (1.5) when $k \geqq n$ and (1.6) when $k < n$ (as suggested by Lieberman and Owen). The examples considered later in the paper suggest that the accuracy obtained using the binomial approximation is sufficient for practical purposes.

If sample sizes are larger than 150, it is usually convenient to use the Poisson approximation to the binomial. Two good tables, which together contain about all the probabilities that would ever be needed, are the ones prepared by General Electric [2] and Molina [6].

If more accuracy is desired than can be obtained from the approximations (which is unlikely in most applications), then a high speed computer can be used to obtain a solution by following the same procedure demonstrated in the examples.

2. THE SINGLE SAMPLE CASE

A sample of size n is selected at random from a lot of size N. Let X be the number of defectives in the sample. If $x \leqq c$ defective items are found in the sample, these items are replaced by nondefectives and the lot is accepted without further inspection. If $x > c$ the lot is totally inspected and all defective items in the lot are replaced by nondefectives. If the lot contains k defectives, then the operating characteristic is

$$OC = P(N, n, k, c). \tag{2.1}$$

When $k=k_1=Np_1$ we wish to accept the lot with probability at most β_1 so that n and c must satisfy the inequality

$$P(N, n, k_1, c) \leqq \beta_1 \tag{2.2}$$

For a lot containing k items the expected number of items inspected is

$$I_S=n+(N-n)[1-P(N, n, k, c)] \tag{2.3}$$

However, if the process average is p, then k is an assumed value of a random variable and the number of defective items in a sample of size n has an unconditional binomial distribution with parameters n and p. In other words I_S is a conditional expectation, the unconditional expected value being

$$I_S=n+(N-n)E(c+1: n, p) \tag{2.4}$$

Dodge and Romig [1] have minimized (2.4) at $p=\bar{p}$ subject to (2.2) with $\beta_1=0.10$ and have tabulated a number of such sampling plans. The minimum value of I_S will be denoted by $\bar{I}_S$.

Using the standard tables mentioned in section 1 the minimization can be accomplished by trial starting with $c=0$, and increasing c one unit at a time. For each c the minimum n satisfying (2.2) is found and I_S is computed. Calculations cease when the minimum is observed. We will demonstrate by examples.

EXAMPLE 2.1: If $N=50$, $k_1=12$, $\beta_1=0.20$ find the plan which minimizes I_S when $p=\bar{p}=0.06$. Find the OC if $\bar{k}=N\bar{p}=3$.

SOLUTION: Condition (2.2) becomes $P(50, n, 12, c) \leqq 0.20$. With the Lieberman and Owen [5] table we verify that if $c=0$, $n \geqq 6$, if $c=1$, $n \geqq 16$, if $c=3$, $n \geqq 20$, etc. Obviously we choose the minimum n in each interval. Using the Ordnance Corps [7] (or Harvard [4]) and the Lieberman and Owen tables, we find:

if $c=0$ and $n=6$, $I_S=6+44E(1: 6, 0.06)=6+44(0.310130)=19.65$;
if $c=1$ and $n=11$, $I_S=11+39E(2: 11, 0.06)=11+39(0.138216)=16.39$;
if $c=2$ and $n=16$, $I_S=16+34E(3: 16, 0.06)=16+34(0.067280)=18.29$.

Further calculations are obviously unnecessary and the plan which minimizes I_S is $n=11$, $c=1$, with $\bar{I}_S=16.39$.

The OC at $\bar{k}=3$ is $OC=P(50, 11, 3, 1)=0.882143$.

EXAMPLE 2.2: If $N=1000$, $k_1=100(p_1=0.10)$, $\beta_1=0.10$ find the plan which minimizes the average amount of inspection if $\bar{p}=0.02$. Find the OC when $k=\bar{k}=N\bar{p}$.

SOLUTION: With $N=1,000$, we are out of the range of the Lieberman-Owen hypergeometric table, and we use the approximation

$$OC=P(1000, n, 100, c) \cong 1-E(c+1: n, 0.10)$$

(with $n>100$ approximation (1.6) is slightly better) so that (2.2) becomes

$$E(c+1: n, 0.10) \geqq 0.90$$

With the Ordnance Corps [7] table, we verify that if $c=0$, $n \geqq 22$, if $c=1$, $n \geqq 38$, if $c=2$, $n \geqq 52$, if $c=3$, $n \geqq 65$, if $c=4$, $n \geqq 78$, if $c=5$, $n \geqq 91$, if $c=6$, $n \geqq 104$, if $c=7$, $n \geqq 116$, etc. We get with

$c=0,\ n=20 \quad I_S=20+980E(1;\ 20,\ 0.02)=20+980(0.33239)=345.7$

$c=1,\ n=38 \quad I_S=38+962E(2;\ 38,\ 0.02)=38+962(0.17603)=207.3$

$c=2,\ n=52 \quad I_S=52+948E(3;\ 52,\ 0.02)=52+948(0.08593)=133.5$

$c=3,\ n=65 \quad I_S=65+935E(4;\ 65,\ 0.02)=65+935(0.04138)=103.7$

$c=4,\ n=78 \quad I_S=78+922E(5;\ 78,\ 0.02)=78+922(0.02028)=96.7$

$c=5,\ n=91 \quad I_S=91+909E(6;\ 91,\ 0.02)=91+909(0.01006)=100.9$

Obviously further calculations are unnecessary and the desired plan is $n=78$ and $c=4$.

Dodge and Romig [1] give $n=65$, $c=3$, but their plan is designed to cover intervals on both N and $\bar{p}$.

The OC at $k=\bar{k}=1000(0.02)=20$ is $OC\cong 1-E(5;\ 78,\ 0.02)=0.97972$.

Hald [3] has derived asymptotic formulas which, together with some auxiliary tables, can be used to obtain sampling plans of the type we have considered. Considerable calculation seems to be required. His paper has one numerical example which we will now work for comparison of results.

EXAMPLE 2.3: If $N=280$, $p_1=0.10(k_1=28)$, $\beta_1=0.10$, find the plan which minimizes the average amount of inspection if $\bar{p}=0.045$.

SOLUTION: We need

$$OC=P(280,\ n,\ 28,\ c)\cong 1-E\left(c+1;\ 28,\ \frac{n}{280}\right)\leqq 0.10$$

or

$$E\left(c+1;\ 28,\ \frac{n}{280}\right)\geqq 0.90$$

and

$$I_S=n+(280-n)E(c+1;\ n,\ 0.045).$$

Without interpolating on p in the binomial table we find if

$c=0,\ n/280\geqq 0.08,\ n\geqq\ 26,\ I_S=\ 25+254(0.69794)=203$

$c=1,\ n/280\geqq 0.14,\ n\geqq\ 40,\ I_S=\ 40+240(0.54265)=170$

$c=2,\ n/280\geqq 0.18,\ n\geqq\ 51,\ I_S=\ 51+229(0.40442)=144$

$c=3,\ n/280\geqq 0.23,\ n\geqq\ 65,\ I_S=\ 65+215(0.33554)=137$

$c=4,\ n/280\geqq 0.27,\ n\geqq\ 76,\ I_S=\ 76+204(0.25699)=128$

$c=5,\ n/280\geqq 0.31,\ n\geqq\ 87,\ I_S=\ 87+193(0.19784)=125$

$c=6,\ n/280\geqq 0.35,\ n\geqq\ 98,\ I_S=\ 98+182(0.15299)=126$

$c=7,\ n/280\geqq 0.39,\ n\geqq 110,\ I_S=110+170(0.\ 1283)=132$

where $E(c+1;\ n,\ 0.045)$ was found from the Weintraub [8] table except for $n=116$ for which the Poisson approximation was used. Recomputing the three smallest I_S's using linear interpolation in the binomial table yields with

$$c=4,\ n/280\geqq 0.2655,\ n\geqq 75,\ I_S=75+205(0.24845)=125.9$$

$c=5,\ np/280 \geqq 0.3065,\ n \geqq 86,\ I_S=86+194(0.19090)=123.0$
$c=6,\ np/280 \geqq 0.3460,\ n \geqq 97,\ I_S=97+183(0.14740)=124.0$

The plan with minimum I_S is $n=86$, $c=5$ with $\bar{I}_S=123.0$. Hald gives $n=84$, $c=5$, $\bar{I}_S=119.6$ but his Consumer's Risk is slightly larger than 0.10 while ours (within the limits of the approximation) has the Consumer's Risk slightly less than 0.10.

The average outgoing quality for the single sample case is

$$AOQ=\left(1-\frac{n}{N}\right)p[1-E(c+1;\ n,\ p)] \tag{2.5}$$

The maximum of (2.5) taken over p is called the average outgoing quality limit (*AOQL*). Dodge and Romig [1, pp. 37–39] describe a method of approximating *AOQL*. We observe that *AOQL* can also be found by trial using Weintraub's [8] table. For Example 2.2 in which $N=1000$, $n=78$, $c=2$ we find

if $p=0.045$	$AOQ=0.922(0.045)(0.72575)=0.03011$
$p=0.046$	$AOQ=0.922(0.046)(0.71065)=0.03014$
$p=0.047$	$AOQ=0.922(0.047)(0.69538)=0.03013$

so that $AOQL=0.030$. The Dodge-Romig solution also gives $AOQL=0.030$, occurring at $p=0.0467$.

3. THE DOUBLE SAMPLE CASE

A sample of size n_1 is selected at random from a lot of size N. Let X_1 be the number of defective items in the sample. If $x_1 \leqq c_1$ defective items are found in the sample, these items are replaced by non-defectives and the lot is accepted without further inspection. If $c_1 < x_1 \leqq c_2$ a second sample of size n_2 is selected at random from the remaining $N-n_1$ items and X_2, the number of defective items in the second sample, is observed. If $c_1 < x_1+x_2 \leqq c_2$ the lot is accepted without further inspection, but all defective items found in both samples are replaced by good ones. If either $x_1 > c_2$ or $c_1 < x_1 \leqq c_2$, and $x_1+x_2 \geqq c_2$ the lot is totally inspected and all defective items in the lot are replaced by non-defectives. If the lot contains k defectives, then the operating characteristic is

$$OC=H(k;\ N,\ n_1,\ n_2,\ c_1,\ c_2) \tag{3.1}$$

$$=P(N,\ n_1,\ k,\ c_1)+\sum_{j=1}^{c_2-c_1} p(N,\ n_1,\ k,\ c_1+j)\,P(N-n_1,\ n_2,\ k-c_1-j,\ c_2-c_1-j).$$

The counterparts of (2.3) and (2.4) are

$$J_D=n_1+n_2[1-P(N,\ n_1,\ k,\ c_1)]+(N-n_1-n_2)[1-H(k;\ N,\ n_1,\ n_2,\ c_1,\ c_2)] \tag{3.2}$$

and

$$I_D=n_1+n_2\,E(d_1;\ n_1,\ p)+(N-n_1-n_2)\,K(p;\ n_1,\ n_2,\ d_1,\ d_2) \tag{3.3}$$

where $d_1 = c_1 + 1$, $d_2 = c_2 + 1$ and

$$K(p; n_1, n_2, d_1, d_2) = E(d_2; n_1, p) + \sum_{j=0}^{d_2-d_1-1} b(d_1+j; n_1, p)\, E(d_2-d_1-j; n_2, p). \tag{3.4}$$

Now we wish to minimize I_D with $p = \bar{p}$ subject to the condition that the OC at $k = k_1$ be no greater than β_1. That is, we require

$$H(k_1; N, n_1, n_2, c_1, c_2) \leqq \beta_1. \tag{3.5}$$

The minimum value of (3.3) at $p = \bar{p}$ will be denoted by $\bar{I}_D$. If $N > 50$ so that it is not practical to use the table of Lieberman and Owen [5], then we will use binomial approximations for hypergeometric sums, power instead of OC, and condition (3.5) is replaced by

$$K(p_1; n_1, n_2, d_1, d_2) \geqq 1 - \beta_1, \tag{3.6}$$

where $p_1 = k_1/N$.

First we make some general observations.

1. For given n_1, c_1, c_2 we select n_2 as small as possible so as to satisfy (3.5). Denote this choice of n_2 by n_{2L}. Although it is intuitively obvious that larger n_2 just make I_D larger, this is true because $K(p; n_1, n_2, d_1, d_2) < E(d_1; n_1, p)$ a result which follows from (3.4) and the fact that $E(d_2 - d_1 - j; n_2, p) \leqq 1$.

2. Because of the fact that $OC \leqq \beta_1$ when $k = k_1$ it is necessary that

$$P(N, n_1, k_1, c_1) \leqq \beta_1. \tag{3.7}$$

When the binomial approximation is used (3.7) becomes

$$E(d_1; n_1, p_1) \geqq 1 - \beta_1. \tag{3.8}$$

These inequalities provide a lower bound on n_1 giving $n_1 \geqq n_{1l}$. Also if

$$E(d_2; n_1, p_1) \geqq 1 - \beta_1, \tag{3.9}$$

then $K(p_1; n_1, n_2, d_1, d_2) \geqq 1 - \beta_1$ for every $n_2 \geqq 0$. Hence if n_{1l} is the smallest n_1 to satisfy (3.9), then we need consider only $n_1 \leqq n_{1l}$. The hypergeometric condition corresponding to (3.9) is

$$P(N, n_1, k_1, c_2) \leqq \beta_1. \tag{3.10}$$

We observe that n_{1l} is the single sample solution satisfying (2.2) with $n = n_{1l}$, $c = c_2$.

3. Only if $n_1 < \bar{I}_s$ do we need to consider a plan since otherwise $I_D > \bar{I}_s$ and the objective of double

sampling (to reduce average total inspection) would be defeated.

4. For chosen c_1, c_2 we must have $n_1+n_2 \geqq n_{1U}$. To see this assume that the converse is true, that is, there exist $n_1+n_2 < n_{1U}$. Then the power at $k=k_1$ is made $\geqq 1-\beta_1$ by taking n_1 observations all of the time and n_2 observations part of the time. The power is not decreased if the second sample is taken with probability 1. But this means that a single sample plan with the given c_2 exists with $n=n_1+n_2 < n_{1U}$, contrary to the definition of n_{1U}.

5. As n_1 increases n_1+n_2 is nonincreasing and has as its minimum value n_{1U} (attainable when $n_1=n_{1U}$, $n_2=0$). This sum may be considerably greater than n_{1U} when $n_1=n_{1L}$, but gets close to n_{1U} after n_1 has been increased by relatively few units. This is explained by observing that when $n_1=n_{1L}$, $E(d_1; n_1, p)$, which is greater than the power, is very nearly $1-\beta_1$ and to satisfy (3.6) the terms $E(d_2-d_1-j; n_2, p)$ must be large so that n_2 is large. As n_1 increases the difference between power and $E(d_1; n_1, p)$ grows at a relatively rapid pace permitting $E(d_2-d_1-j; n_2, p)$ and n_2 to be much smaller.

In a numerical problem we suggest the following steps:

1. Calculate $\bar{I}_S$ since, as we have already mentioned, it is not necessary to consider plans for which $I_p > \bar{I}_S$.

2. With $c_1=0$ determine n_{1L} from (3.7) or (3.8).

(a) With $c_2=1$

(1) Find n_{1U} from (3.9) or (3.10).

(2) By trial (using the fact that $n_1+n_2 \geqq n_{1U}$) find the minimum value of n_2, say n_{2L}, which satisfies (3.5) or (3.6).

(3) With n_{1L}, n_{2L} find I_p.

(b) Repeat (a) with $c_2=2$. As a first guess for the new n_{2L} increase the old n_{2L} by the same amount n_{1U} has increased.

(c) Repeat (a) with $c_2=3$.

etc.

Terminate when it is obvious that I_p must increase with further increase in c_2.

3. Repeat Step 2 with n_{1L} replaced by $n_{1L}+1$. Then repeat Step 2 with n_{1L} replaced by $n_{1L}+2$, etc., terminating when it is obvious that a minimum has been found for each c_2 for which it has been necessary to consider $c_1=0$.

4. Repeat Steps 2 and 3 with $c_1=1$, $c_1=2$, then $c_1=3$, etc., terminating when values of I_p get too large. This happens at worst when $n_1 > \bar{I}_S$.

5. By observation select $\bar{I}_p$.

Although the procedure outlined in the previous paragraph may require a number of calculations, it goes rather quickly using a desk calculator which has cumulative multiplication. When using the hypergeometric tables it is probably advisable to copy down all figures before going to the calculator (because of the format of the table). In the binomial case it is advisable to copy down $E(d_2; n_1, p)$ and the $b(d_1+j; n_1, p)$. However, the $E(d_2-d_1-j; n_2, p)$ may be transferred directly from the binomial table to the calculator and need not be copied. The major advantage of proceeding as suggested in the previous paragraph is that in the calculation of power or *OC* all previous $b(d_1+j; n_1, p)$ or $p(N, n_1, k_1, c_1+j)$ are used plus one more as d_2 or c_2 is increased by a unit. We now consider examples.

EXAMPLE 3.1: If $N=50$, $k_1=12$, $\beta_1=0.20$, find the double sampling plan which minimizes I_b when $p=\bar{p}=0.06$. Find the OC for the required plan when $\bar{k}=N\bar{p}=3$.

SOLUTION: In Example 2.1 we already found that $\bar{I}_s=16.39$. Also we had that if $c=0$, $n\geqq 6$, if $c=1$, $n\geqq 11$, if $c=2$, $n\geqq 16$, if $c=3$, $n\geqq 20$.

We begin by selecting $c_1=0$. Then possible values for c_2 are 1, 2, 3, 4, etc., and the OC is

$$H(k;50,n_1,n_2,0,c_2)=P(50,n_1,k,0)+\sum_{j=1}^{c_2} p(50,n_1,k,j)\,P(50-n_1,n_2,k-j,c_2-j)$$

Condition (3.7) is $P(50,n_1,12,0)\leqq 0.20$ which requires $n_1\geqq 6=n_{1t}$.

If $c_2=1$ condition (3.10) is $P(50,n_1,12,1)\leqq 0.20$ so $n_{1t}=11$. With $n_1=6$, $k_1=12$ the OC is

$$\begin{aligned}H(12;50,6,n_2,0,1)&=P(50,6,12,0)+p(50,6,12,1)\,P(44,n_2,11,0)\\&=0.173729+0.379046\,P(44,n_2,11,0)\end{aligned}$$

By trial we find $H(12;50,6,9,0,1)=0.194350$, $H(12;50,6,8,0,1)=0.203423$ so that $n_1=6$, $n_2=9$, $c_1=0$, $c_2=1$ is a possible plan. Then

$$\begin{aligned}K(.06;6,9,1,2)&=E(2;6,0.06)+b(1;6,0.06)\,E(1;9,0.06)\\&=0.01592+(0.26421)\,(0.42701)=0.15874\end{aligned}$$

and when $\bar{p}=0.06$

$$\begin{aligned}I_b&=6+9\,E(1;6,0.06)+35\,K(0.06;6,9,1,2)\\&=6+9(0.31013)+35(0.15874)=14.55\end{aligned}$$

Next we take $c_2=2$ with $c_1=0$, $n_1=6$. Now (3.10) is $P(50,n_1,12,2)\leqq 0.20$ and $n_{1t}=16$. The OC with $k_1=12$ is

$$\begin{aligned}H(12;50,6,n_2,0,2)&=P(50,6,12,0)+p(50,6,12,1)\,P(44,n_2,11,1)\\&\quad+p(50,6,12,2)\,P(44,n_2,10,0)\\&=0.173729+0.379046\,P(44,n_2,11,1)\\&\quad+0.306581\,P(44,n_2,10,0).\end{aligned}$$

By trial we find (a good first guess is $n_2=9+5=14$) $H(12;50,6,15,0,2)=0.192763$, $H(12;50,6,14,0,2)=0.200929$ so that $n_1=6$, $n_2=15$, $c_1=0$, $c_2=2$ is a possible plan. Then

$$\begin{aligned}K(0.06;6,15,1,3)&=E(3;6,0.06)+b(1;6,0.06)\,E(2;15,0.06)\\&\quad+b(2;6,0.06)\,E(1;15,0.06)\\&=0.00376+(0.26421)\,(0.22624)\\&\quad+(0.04226)\,(0.60741)=0.08909\end{aligned}$$

and when $\bar{p}=0.06$

$$\begin{aligned}I_b&=6+15\,E(1;6,0.06)+29\,K(0.06;6,15,0,2)\\&=6+15(0.31013)+29(0.08909)=13.24\end{aligned}$$

We next take $c_2=3$ with $c_1=0$, $n_1=6$. Now $n_1+n_2\geqq n_{1t}=20$ and the OC with $k_1=12$ is

$$\begin{aligned}H(12;50,6,n_2,0,3)&=P(50,6,12,0)+p(50,6,12,1)\,P(44,n_2,11,2)\\&\quad+p(50,6,12,2)\,P(44,n_2,10,1)\\&\quad+p(50,6,12,3)\,P(44,n_2,9,0)\end{aligned}$$

$$= 0.173729 + (0.379046)\, P(44, n_2, 11, 2) + (0.306581)\, P(44, n_2, 10, 1) + (0.116793)\, P(44, n_2, 9, 0)$$

By trial we find (a good first guess is $n_2 = 15+4 = 19$) $H(12; 50, 6, 19, 0, 3) = 0.199822$, $H(12; 50, 6, 18, 0, 3) = 0.210583$ so that $n_1 = 6$, $n_2 = 19$, $c_1 = 0$, $c_2 = 3$ is a possible plan. Then

$$K(0.06; 6, 19, 1, 4) = E(4; 6, 0.06) + b(1; 6, 0.06)\, E(3; 19, 0.06) + b(2; 6, 0.06)\, E(2; 19, 0.06) + b(3; 6, 0.06)\, E(1; 19, 0.06)$$
$$= 0.00018 + (0.26421)(0.10207) + (0.04226)(0.31709) + (0.00358)(0.69138) = 0.04302$$

and when $\bar{p} = 0.06$

$$I_b = 6 + 19(0.31031) + 25(0.04302) = 12.97.$$

Similarly, with $c_1 = 0$, $c_2 = 4$, $n_1 = 6$, we find $n_2 = 11$ and $I_b = 13.76$. Examination of the results and the terms of I_b indicate that there is no point in continuing the calculations with $n_1 = 6$, $c_1 = 0$.

We next repeat all the above steps with $c_1 = 0$, $n_1 = 7$, then with $c_1 = 0$, $n_1 = 8$, etc., continuing until it is obvious that a minimum has been found for each value of c_2 which it has been necessary to consider. With $c_1 = 0$ we get the following (n_1, n_2) and I_b:

$c_2 = 1$	$c_2 = 2$	$c_2 = 3$	$c_2 = 4$
(6, 9), 14.35	(6, 15), 13.24	(6, 19), 12.97	(6, 24), 13.76
(7, 5), 14.04	(7, 11), 12.46	(7, 16), 13.63	(7, 20), 14.42
(8, 4), 15.16	(8, 9), 13.82	(8, 14), 14.44	(8, 18), 15.40

Now we repeat all the steps with $c_1 = 1$. This time c_2 can take on values 2, 3, 4, etc. We get the following (n_1, n_2) and I_b:

$c_2 = 2$	$c_2 = 3$	$c_2 = 4$
(11, 9), 14.72	(11, 15), 14.08	(11, 19), 14.04
(12, 5), 14.98	(12, 10), 14.49	(12, 15), 14.71
(13, 3), 15.70	(13, 8), 15.53	(13, 13), 15.72

It is not necessary to consider higher values of c_2 since the first two terms of I_b are at least $11 + 19\, E(2; 11, 0.06) = 11 + 19(0.13822) = 13.63$ which already exceeds some I_b already obtained.

We do not consider $c_1 = 2$ since now $n_1 \geq 16$ and $I_b > 16$. Hence calculations are terminated.

We observe that the desired plan is $n_1 = 7$, $n_2 = 11$, $c_1 = 0$, $c_2 = 2$ and $\bar{I}_b = 12.46$. On the average this is $16.39 - 12.46 = 3.93$ units less than with single sampling. No comparison with Dodge-Romig [1] is possible since $\beta_c = 0.20$ is not an entry in their table.

The OC for this plan when $k = 3$ is

$$H(3: 50, 7, 11, 0.2) = P(50, 7, 3, 0) + p(50, 7, 3, 1)\, P(43, 11, 2, 1) + p(50, 7, 3, 2)\, P(43, 11, 1, 0)$$
$$= 0.629643 + 0.322500(0.939092) + 0.046071(0.744186)$$
$$= 0.966786.$$

Recall that for single sampling we had $OC = 0.882143$.

EXAMPLE 3.2: If $N = 1000$, $k_1 = 100 (p_1 = 0.10)$, $\beta_1 = 0.10$ find the double sampling plan which minimizes the average amount of inspection of $\bar{p} = 0.02$. Find the OC when $k = \bar{k} = N\bar{p}$.

SOLUTION: In Example 2.2 we already found that $\bar{I}_s = 96.7$. Also we had that if $c = 0$, $n \geqq 22$, if $c = 1$, $n \geqq 38$, if $c = 2$, $n \geqq 52$, if $c = 3$, $n \geqq 65$, if $c = 4$, $n \geqq 78$, etc.

We will omit the calculations and results for $c_1 = 0$ and $c_1 = 2$, demonstrating the procedure with $c_1 = 1$, the value which yields $\bar{I}_D$. Now c_2 can be 2, 3, 4, etc. Condition (3.8) is $E(2; n_1, 0.10) \geqq 0.90$ which requires that $n_1 \geqq 38$.

If $c_2 = 2$ (or $d_2 = 3$) (3.9) yields $n_{12} = 52$ and we must have $n_1 + n_2 \geqq 52$. With $n_1 = 38$ the power is

$$K(p; 38, n_2, 2, 3) = E(3; 38, p) + b(2; 38, p) E(1; n_2, p)$$

and

$$K(0.10; 38, n_2, 2, 3) = E(3; 38, 0.10) + b(2; 38, 0.10) E(1; n_2, 0.10)$$
$$= 0.74633 + 0.15837 E(1; n_2, 0.10).$$

By trial we find

$$K(0.10; 38, 34, 2, 3) = 0.90030, K(0.10; 38, 33, 2, 3) = 0.89981$$

so that $n_1 = 38$, $n_2 = 34$, $c_1 = 1$, $c_2 = 2$ is a possible plan. Then

$$K(0.02; 38, 34, 2, 3) = E(3; 38, 0.02) + b(2; 38, 0.02) E(1; 34, 0.02)$$
$$= 0.04015 + (0.13588)(0.49686) = 0.10766$$

and when $\bar{p} = 0.02$

$$I_D = 38 + 34 E(2; 38, 0.10) + 972 K(0.02; 38, 34, 2, 3)$$
$$= 38 + 34(0.17603) + 972(0.10766) = 143.89.$$

We next take $c_2 = 3$ with $c_1 = 1$, $n_1 = 38$. Now $n_1 + n_2 \geqq 65$ and the power is

$$K(p; 38, n_2, 2, 4) = E(4; 38, p) + b(2; 38, p) E(2; n_2, p) + b(3; 38, p) E(1; n_2, p)$$

and

$$K(0.10; 38, n_2, 2, 4) = 0.53516 + (0.15837) E(2; n_2, 0.10) + (0.21117) E(1; n_2, 0.10)$$

By trial we find (we might first guess $n_2 = 47$)

$$K(0.10; 38, 54, 2, 4) = 0.90024, K(0.10; 38, 53, 2, 4) = 0.89980$$

so that $n_1 = 38$, $n_2 = 54$, $c_1 = 1$, $c_2 = 3$ is a possible plan. Then

$$K(0.02; 38, 54, 2, 4) = E(4; 38, 0.02) + b(2; 38, 0.02) E(2; 54, 0.02) + b(3; 38, 0.02) E(1; 54, 0.02)$$
$$= 0.00687 + (0.13588)(0.29393) + (0.03327)(0.66410) = 0.06890$$

and when $\bar{p}=0.02$

$$I_p=38+54E(2:38.0.10)+908K(0.02:38.54.2.4)$$
$$=38+54.0.17603)+908(0.06890)=100.07.$$

We next take $c_2=4$ with $c_1=1$. $n_1=38$. Now $n_1+n_2\geq 78$ and the power is

$$K(p:38.n_2.2.5)=E(5:38.p)+b(2:38.p)E(3:n_2.p)$$
$$+b(3:38.p)E(2:n_2.p)$$
$$+b(4:38.p)E(1:n_2.p)$$

and

$$K(0.10:38.n_2.2.5)=0.32986+(0.15837)E(3:n_2.0.10)$$
$$+(0.21117)E(2:n_2.0.10)$$
$$+(0.20530)E(1:n_2.0.10)$$

By trial we find (we might guess $n_2=67$)

$$K(0.10:38.72.2.5)=0.90037.K(0.10:38.71.2.5)=0.89999$$

so that $n_1=38$. $n_2=72$. $c_1=1$. $c_2=4$ is a possible plan. Then

$$K(0.02:38.72.2.5)=0.00093+(0.13588)(0.17484)$$
$$+(0.03327)(0.42341)$$
$$+(0.00594)(0.76651)=0.04333$$

and when $\bar{p}=0.02$

$$I_p=38+72(0.17603)+890(0.04333)=93.14$$

Similarly with $c_2=5$. we find:

$$I_p=38+88(0.17603)+874(0.02615)=76.35$$

With $c_2=6$ we get

$$I_p=38+103(0.17603)+859(0.01533)=69.30$$

With $c_2=7$ we get

$$I_p=38+119(0.17603)+843(0.00922)=66.72$$

With $c_2=8$ we get

$$I_p=38+133(0.17603)+829(0.00522)=65.74$$

It appears that if c_2 is increased to 9 the increase in the second term of I_p will be roughly the same as the decrease of the third term. Thus. for the moment at least. further calculations with $n_1=38$ seem unnecessary.

Next we repeat all the above steps for $c_1=1$ with $n_1=39$. then $n_1=40$. etc.. until it is obvious that

TABLE 1. $(n_1.\ n_2)$ and I_p for $c_1=1$

$c_2=2$	$c_2=3$	$c_2=4$	$c_2=5$	$c_2=6$	$c_2=7$	$c_2=8$
(38, 34, 143.09	(38, 54, 100.07	(38, 72, 93.14	(38, 88, 76.35	(38, 103, 69.30	(38, 119, 66.72	(38, 133, 65.74
(39, 24, 134.09	(39, 43, 99.98	(39, 59, 79.86	(39, 75, 69.91	(39, 89, 61.69	(39, 104, 63.24	(39, 119, 63.71
(40, 19, 130.02	(40, 37, 95.41	(40, 53, 76.90	(40, 68, 63.71	(40, 82, 63.78	(40, 96, 62.13	(40, 110, 63.17
(41, 16, 128.82	(41, 33, 92.64	(41, 49, 75.68	(41, 63, 66.40	(41, 77, 63.11	(41, 92, 62.97	(41, 105, 63.02
(42, 14, 129.21	(42, 30, 92.02	(42, 45, 75.01	(42, 60, 66.42	(42, 74, 63.66	(42, 87, 63.12	(42, 101, 64.44
(43, 13, 126.75	(43, 28, 92.19	(43, 43, 74.82	(43, 57, 66.81	(43, 71, 64.18	(43, 84, 64.26	(43, 97, 65.33
(44, 10, 128.28	(44, 26, 92.27	(44, 40, 74.22	(44, 54, 67.30	(44, 68, 64.64	(44, 81, 64.66	(44, 94, 66.10
(45, 8, 128.07	(45, 24, 92.17	(45, 38, 74.55	(45, 52, 67.50	(45, 66, 65.33	(45, 79, 65.59	(45, 92, 67.29
(46, 7, 129.82	(46, 22, 91.71					
(47, 5, 128.39	(47, 21, 93.01					

we have a minimum for each c_2. The results in Table 1 are obtained. From the table it is observed that $\bar{I}_D = 62.43$ (given that the minimum does not occur with $c_1 = 0$ or $c_1 = 2$) and the desired plan is $n_1 = 40$, $n_2 = 96$, $c_1 = 1$, $c_2 = 7$. We note that it is unnecessary to consider $c_1 = 3$ (or greater) since the condition $E(4; n_1, 0.10) \geq 0.90$ requires $n_1 \geq 65$ and we have already found a number of plans with $I_D < 65$.

The OC at $\bar{p} = 0.02$ for the plan which minimizes I_D has value 0.99649. Recall that for the single sample plan of Example 2.2 we had 0.97972.

Dodge and Romig give for the solution to our problem $n_1 = 28$, $n_2 = 72$, $c_1 = 0$, $c_2 = 5$ for which $I_D = 70.89$. $K(0.10; 28, 72, 1, 6) = 0.904$.

The average outgoing quality for the double sample case can be written in various forms, but perhaps the one most convenient for use with tables is

$$AOQ = \frac{n_2}{N} p[1 - E(d_1; n_1, p)] + \left(1 - \frac{n_1 + n_2}{N}\right) p[1 - K(p; n_1, n_2, d_1, d_2)]. \tag{3.11}$$

Again the $AOQL$, the maximum taken over p, can be found by trial using the Weintraub [8] table. For the plan found in Example 3.2 which had $\bar{I}_D = 62.43$ we get

$$\text{if } p = 0.046 \quad AOQ = (0.096)(0.046)(0.99959) + (0.864)(0.046)(0.76707) = 0.03490$$
$$p = 0.047 \quad AOQ = (0.096)(0.047)(0.99953) + (0.864)(0.047)(0.75139) = 0.03502$$
$$p = 0.048 \quad AOQ = (0.096)(0.048)(0.99946) + (0.864)(0.048)(0.73338) = 0.03482$$

so that $AOQL = 0.035$. The $AOQL$ for the corresponding single sample case, found at the end of Section 2, was 0.030. Intuitively we might expect a larger $AOQL$ for a plan which on the average requires less inspection.

REFERENCES

[1] Dodge, Harold F., and Harry G. Romig, *Sampling Inspection Tables Single and Double Sampling* (John Wiley and Sons, Inc., New York, New York, 1959) (2nd ed.).

[2] General Electric Company, *Tables of Poisson Distributions: Individual and Cumulative Terms* (D. Van Nostrand, Princeton, New Jersey, 1962).

[3] Hald, Anders, "Some Limit Theorems for the Dodge-Romig LTPD Single Sampling Inspection Plans," Technometrics, 4, 497–513 (1962).

[4] Harvard University Computation Laboratory, *Tables of the Cumulative Binomial Probability Distribution* (Harvard University Press, Cambridge, Mass., 1955).

[5] Lieberman, G. J. and D. B. Owen, *Tables of the Hypergeometric Probability Distribution* (Stanford University Press, Stanford, Calif., 1961).

[6] Molina, E. C, *Poisson's Binomial Exponential Limit* (D. Van Nostrand, Princeton, New Jersey, 1949).

[7] U.S. Army Ordnance Corps, *Tables of Cumulative Binomial Probabilities*, ORDP20–1 (1952). Office of Technical Services, Washington, D.C. (Reprinted Jan. 1971 as Army Materiel Command Pamphlet No. 706–109.)

[8] Weintraub, Sol, *Tables of the Cumulative Binomial Probability Distribution for Small Values of p* (The Free Press, New York, New York, 1963).

A MODEL FOR MANPOWER PRODUCTIVITY DURING ORGANIZATION GROWTH*

John G. Rau

Ultrasystems, Inc.
Newport Beach, California

A mathematical model is developed that enables organization and manpower planners to quantify the inefficiencies involved in rapid buildups of organizations, such as is frequently found in the aerospace industry shortly after the award of a major contract. Consideration is given to the time required to train, indoctrinate, and familiarize new workers with their jobs and the general program aspects. Once trained, workers are assumed to be productive. If the ratio of untrained to trained workers exceeds a critical value, called the buildup threshold, then the performance of the trained workers is degraded to the extent that they are no longer 100 percent efficient until this ratio returns to a value less than the threshold. The model is sufficiently general to consider an arbitrary manpower plan with more than one peak or valley. The model outputs are functions of real time and consist of the fraction of the total labor force which is productive, the fraction of the total labor units expended for nonproductive effort, the cumulative labor costs for productive effort, and the cumulative labor cost for all effort.

I. INTRODUCTION

During the buildup phase of a program (or, equivalently, during the initial growth period of an organization) new people are assimilated into the program team. These people bring a broad variety of skills, experience, education and training to the organization and are required to support the program in various functional areas such as engineering, tooling, manufacturing, quality control, and logistics. As a result, natural questions arise as to what the buildup and builddown rates should be for these types of workers, what the size of the initial labor force should be, when the buildup should commence and end, and finally when the builddown should begin and end. Typically, management is interested in whether or not there exists an optimal plan for building an organization and, furthermore, what are the alternative criteria which can be used to determine an optimal plan.

The major difficulty in formulating criteria and determining an optimal buildup plan lies in defining the variables which need to be considered, determining how to measure them in a practical way, and discovering what relationships, if any, exist between them. These are in essence three basic steps in developing a mathematical model. In many cases, in order to obtain acceptance by management, the model must be conceptually simple, involve variables which are easily understood and in some sense readily measurable, and yield results which are explainable and potentially useful for planning and forecasting purposes. We shall attempt to accomplish these objectives in the ensuing development and shall show how with a rather small and simple set of variables, one can obtain considerable insight into the nature of the buildup and builddown process.

* Based on research (see Ref. [6]) by the author while with North American Rockwell Corporation, Anaheim, California.

II. LEARNING AND WORKER PRODUCTIVITY

When workers are assimilated into the program organization, they must be trained and indoctrinated into the company way of doing things. This requires, among other things, explanation of company policies and organization structure, the products to be developed and their technical specifications, and, perhaps, training in the use of special tools and techniques which are required in performance of the individual's work assignment. All of this takes time and, from the point of view employee contribution, can be regarded as a "nonproductive" period. This period will be referred to as the learning time, denoted by L, and the basic assumption of the model is that all personnel are "productive" after they have been on the program for L time units. The learning time corresponds to that which Purkiss [5, p. 5] refers to as the time period to train a man. It is not unreasonable to expect that the value of L would depend upon the type of individual hired, such as technician, engineer, tool and die maker, machinist, clerk, and quality control inspector, or even upon the number of years of experience both within and outside the company and educational degrees obtained.

Factors affecting the learning time of a worker can be expected to include the following:

(1) The time spent in general orientation training concerning such subjects as organizational structure, objectives and missions; program purpose and objectives; philosophy and inner workings of the customer's operating and support environment; general systems outline (if the purpose of the organization is to develop and design some type of system).

(2) The time spent in technical orientation training concerning the technical concept of the system design and indoctrination at the technical level of the major and minor subsystems

(3) The time spent in the internal company training/retraining, skills development and skills certification programs. Typical examples are as follows:

(a) Engineering training to accelerate the engineers' integration into the working team and to broaden their comprehension of their specific task assignments.

(b) Manufacturing skill training to provide personnel capable of assuming the duties of an assembly work station and of performing the tasks of the station in accordance with program requirements.

(c) Quality control training which encompasses all techniques, processes, and procedures utilized during design, development, manufacturing, inspection, handling, and packaging.

(d) Reliability training of technicians and skilled craftsmen to assure that their skills and knowledge keep up with the advancing technology required to achieve the specified system reliability requirements.

(e) Training in the proper and safe methods of packing, shipping, and storing of the items used in the manufacture, assembly, and test of hardware items.

(f) Supervisory orientation to provide new and experienced supervisors with program philosophy and review of responsibilities.

(4) The time spent after completion of all general and specialized orientation training until the worker is given a specific and well defined task assignment. This factor is quite important and can usually be attributed to poor management planning and supervisory practices.

Two implications of this basic assumption regarding learning time are:

(1) It implies that each individual reaches a productive level after spending L time units on the program, thus ignoring the fact that some new hires never become productive;
and

(2) It ignores the fact that while an individual is being trained and indoctrinated he can make a meaningful contribution to the program and thus, in a sense, be productive even though not perhaps productive at the level for which he was hired.

One way of resolving (1) would be to introduce into the model an attrition rate for each type of worker hired to allow for recognition by company management that the individual is not going to mature as expected and thus his employment is terminated. Consequently, if the buildup rate is U workers per unit time and the attrition rate (i.e., terminated workers per unit time) is A, then the net buildup rate is $U-A$ workers per unit time; hence, one could merely use $U-A$ instead of U as the buildup rate input.

The assumption of zero productivity by a worker during the first L time units of his employment can be described graphically by the step function in Figure 1.

One may argue that (2) is an unrealistic implication of the basic assumption and that productivity is perhaps a piecewise linear function of the form described in Figure 2. If this is the true situation, then this can be resolved by redefining L to take into consideration the area of the shaded triangle which the assumption, as stated, would otherwise ignore. More specifically, if the line segment for real time between O and L has slope m, then we must have $L=1/m$; hence, the area of the triangle is $1/2m$. Therefore, using the interpretation of learning time according to our assumption, we choose the learning time equal to $1/2m$ because then the area of the shaded rectangle in Figure 3 equals the area of the triangle in Figure 2. In this way, the effect is approximately the same.

III. DEVELOPMENT OF THE BUILDUP AND BUILDDOWN PROCESS

Consider a program which begins at time 0, with $w(0)$ workers of which $p(0)$ are initially trained and the remainder $n(0)=w(0)-p(0)$ are nontrained. At time $t>0$, let $w(t)$ be the size of the program

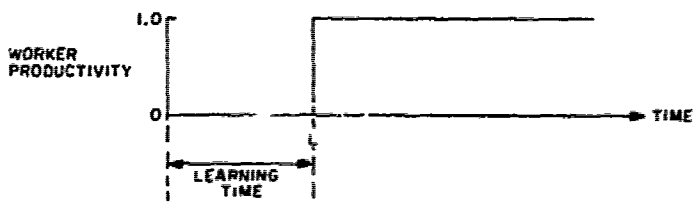


FIGURE 1. Productivity step function

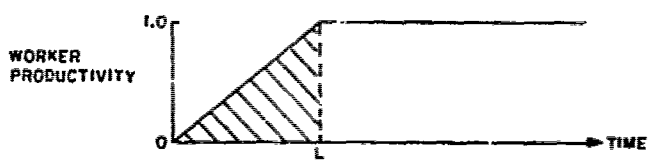


FIGURE 2. Piecewise linear productivity function

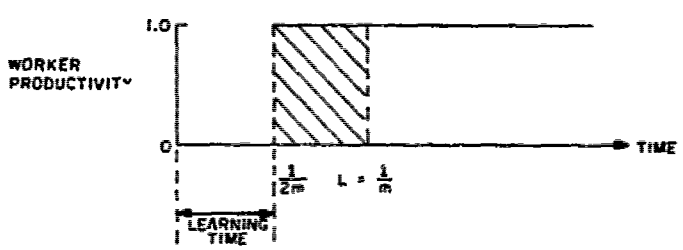


FIGURE 3. Step function approximation

organization, where the units of t could be weeks, months, or quarters. At time t, the number of productive workers (i.e., those who have completed the learning process) is denoted by $p(t)$ and the number of nonproductive workers is denoted by $n(t)$, hence we obtain, since each worker is defined to be either productive or nonproductive,

$$w(t)=n(t)+p(t). \tag{1}$$

Assuming changes at discrete times only,* if $w(t)>w(t-1)$, then the organization has increased at time t by an amount equal to $U(t)=w(t)-w(t-1)$; on the other hand, if $w(t)<w(t-1)$, then the organization has decreased at time t by an amount equal to $D(t)=w(t-1)-w(t)$. In general, we can define the increase in the organization at time $t\geq 1$ relative to time $t-1$ by

$$U(t)=\max\{0, w(t)-w(t-1)\} \tag{2}$$

and the decrease in the organization at time $t\geq 1$ relative to time $t-1$ by

$$D(t)=\max\{0, w(t-1)-w(t)\}. \tag{3}$$

Consequently, we observe that

$$w(t)=w(t-1)+D(t)+U(t). \tag{4}$$

Suppose that, if there is a decrease in the number of workers at time t relative to time $t-1$, the decrease is made in the number of productive workers first, that is, trained workers are removed first. The reason for this assumption is that normally the organization would not decrease in size until all workers were trained (hence, everyone would be productive) and, if nontrained workers were removed first, then some type of seniority rule would have to be assumed such as the removal of the most recently hired workers, then the next most recent, etc. This assumption thus avoids any further assumptions about seniority rules and simplifies the model. Therefore, if $t<L$, where L denotes the worker learning time, none of the initial nontrained workers are yet trained and so the number of productive workers is either zero or equal to

$$p(o)-\sum_{j=1}^{[t]} D(j),$$

where $[t]$ denotes the largest integer not exceeding t; if $t\geq L$, then all the initial workers $w(o)$ are now productive, together with those hired by time $t-L$, (namely,

$$\sum_{j=1}^{[t-L]} U(j)$$

and so the number of productive workers in this case is either zero or equal to

$$w(o)+\sum_{j=1}^{[t-L]} U(j)-\sum_{j=1}^{[t]} D(j).$$

*Since organization size changes typically on a daily, weekly, or monthly basis and not continuously with time.

Consequently, we have for $t \geq 1$

$$p(t) = \begin{cases} \max\{0,\ p(o) - \sum_{j=1}^{|a|} D(j)\} & \text{if } 1 \leq t < L \\ \max\{0,\ w(o) + \sum_{j=1}^{|t-L|} U(j) - \sum_{j=1}^{|a|} D(j)\} & \text{if } L \leq t. \end{cases}$$

The number of nonproductive workers, $n(t)$, is then determined from Equation (1).

IV. SPECIAL CASE OF CONSTANT BUILDUP AND BUILDDOWN RATES

As an illustration of the model as formulated, let us consider the special case in which $U(t)$ is a positive constant whenever it is not zero and, similarly, $D(t)$ is a positive constant whenever it is not zero; that is, we are considering the constant buildup rate and constant builddown rate situation. Let us suppose that the organization begins at time $T_S = 0$ with a basic core, $w(o)$, of workers and once the planning of tasks and training programs are defined, say at time T_I, the buildup is begun at a constant rate U. Upon reaching the peak labor force, say at time T_P, the organization stays at this size until it is deemed feasible and necessary to start laying-off workers, say at time T_D, in which case the organization builds down at a constant rate D to a basic minimal size (this occurs at time T_L, say) sufficient to carry out the remaining tasks and efforts of the program, and then continues until time T_E at which time the program ends.

In this special case, we have for $t \geq 1$

$$w(t) = \begin{cases} w(t-1) & \text{if } t < T_I,\ T_P < t < T_D \text{ or } T_L < t \\ w(t-1) + U & \text{if } T_I \leq t \leq T_P \\ w(t-1) - D & \text{if } T_D \leq t \leq T_L. \end{cases} \tag{6}$$

However, since workers are only added to the program at the discrete times T_I, $T_{I+1}, \ldots, T_P$, the total number of workers hired by time t where $T_I \leq t \leq T_P$ is $U[t - T_I + 1]$. Since workers are only removed from the organization at times T_D, $T_D + 1, \ldots, T_L$, the total number of workers laid off by time t where $T_D \leq t \leq T_L$ is $D[t - T_D + 1]$. Therefore, it follows that

$$w(t) = \begin{cases} 0 & \text{if } t < T_S \\ w(T_S) & \text{if } T_S \leq t < T_I \\ w(T_S) + U[t - T_I + 1] & \text{if } T_I \leq t < T_P \\ w(T_S) + U(T_P - T_I + 1) & \text{if } T_P \leq t < T_D \\ w(T_S) + U(T_P - T_I + 1) - D[t - T_D + 1] & \text{if } T_D \leq t < T_L \\ w(T_S) + U(T_P - T_I + 1) - D(T_L - T_D + 1) & \text{if } T_L \leq t \leq T_E. \end{cases} \tag{7}$$

We shall, for convenience (to avoid enumerating a myriad of cases) assume that

$$L \leq \min\{T_D - T_I,\ T_I - T_P,\ T_P - T_S\}. \tag{8}$$

This assumption means that some of the workers added to the organization during the buildup process have an opportunity to complete the learning phase before the builddown begins (i.e., $T_I + L \leq T_D$), the last group of U workers added to the program at time T_P can complete their learning phase before

the builddown is completed (i.e., $T_P+L \leq T_L$), and the initial number of nonproductive workers, $n(T_S)$, is productive by the time the peak buildup occurs (i.e., $T_S+L \leq T_P$).

To determine the number, $p(t)$, of productive workers at time t, we observe that any nonproductive worker who has been in the organization at least L time units is productive. This means that all workers hired at T_S, T_S+1, ..., $[t-L]$ are productive by time t and if $t \geq T_P+L$, then all workers have completed their learning phase by time t. Consequently, it is a straightforward argument to show that $p(t)$ is given as follows:

CASE: $T_S+L \leq T_t$ and $T_P+L \leq T_D$

$$(9)\qquad p(t)=\begin{cases}0 & \text{if } t<T_S\\ p(T_S) & \text{if } T_S \leq t<T_S+L\\ w(T_S) & \text{if } T_S+L \leq t<T_t+L\\ w(T_S)+U[t-T_t-L+1] & \text{if } T_t+L \leq t<T_P+L\\ w(T_S)+U(T_P-T_t+1) & \text{if } T_P+L \leq t<T_D\\ w(T_S)+U(T_P-T_t+1)-D[t-T_D+1] & \text{if } T_D \leq t<T_L\\ w(T_S)+U(T_P-T_t+1)-D(T_t-T_D+1) & \text{if } T_L \leq t \leq T_E.\end{cases}$$

CASE: $T_S+L \leq T_t$ and $T_D<T_P+L \leq T_L$

$$(10)\qquad p(t)=\begin{cases}0 & \text{if } t<T_S\\ p(T_S) & \text{if } T_S \leq t<T_S+L\\ w(T_S) & \text{if } T_S+L \leq t<T_t+L\\ w(T_S)+U[t-T_t-L+1] & \text{if } T_t+L \leq t<T_D\\ w(T_S)+U[t-T_t-L+1]-D[t-T_D+1] & \text{if } T_D \leq t<T_P+L\\ w(T_S)+U(T_P-T_t+1)-D[t-T_D+1] & \text{if } T_P+L \leq t<T_t\\ w(T_S)+U(T_P-T_t+1)-D(T_t-T_D+1) & \text{if } T_t \leq t \leq T_E.\end{cases}$$

CASE: $T_t<T_S+L \leq T_P$ and $T_P+L \leq T_D$

$$(11)\qquad p(t)=\begin{cases}0 & \text{if } t<T_S\\ p(T_S) & \text{if } T_S \leq t<T_S+L\\ w(T_S)+U[t-T_t-L+1] & \text{if } T_S+L \leq t<T_P+L\\ w(T_S)+U(T_P-T_t+1) & \text{if } T_P+L \leq t<T_D\\ w(T_S)+U(T_P-T_t+1)-D[t-T_D+1] & \text{if } T_D \leq t<T_t\\ w(T_S)+U(T_P-T_t+1)-D(T_t-T_D+1) & \text{if } T_t \leq t \leq T_E.\end{cases}$$

CASE: $T_t<T_S+L \leq T_P$ and $T_D<T_P+L \leq T_L$

$$(12)\qquad p(t)\begin{cases}0 & \text{if } t<T_S\\ p(T_S) & \text{if } T_S \leq t<T_S+L\\ w(T_S)+U[t-T_t-L+1] & \text{if } T_S+L \leq t<T_D\\ w(T_S)+U]t-T_t-L+1]-D[t-T_D+1] & \text{if } T_D \leq t<T_P+L\\ w(T_S)+U(T_P-T_t+1)-D[t-T_D+1] & \text{if } T_P+L \leq t<T_t\\ w(T_S)+U(T_P-T_t+1)-D(T_t-T_D+1) & \text{if } T_t \leq t \leq T_E.\end{cases}$$

Using the relationship $n(t)=w(t)-p(t)$, one can easily derive $n(t)$ for each of the preceding cases.

V. DETERMINATION OF THE LABOR UNIT EXPENDITURE FUNCTIONS

If there are $w(x)$ workers in the organization at time x, then $w(x)dx$ is the amount of effort (man-hours, mandays, manmonths, etc., depending on whether or not x is expressed in hours, days, or months) expended during the interval $(x, x+dx)$; hence, letting $H(t)$ denote the total amount of effort expended by time t, we have

$$H(t)=\begin{cases}0 & \text{if } t<T_s\\ \int_{T_s}^{t} w(x)\,dx & \text{if } T_s\le t\le T_E,\end{cases} \tag{13}$$

where T_s = the start time for the organization. Typically, of course, one would set $T_s=0$, but if one has a mix of workers in an organization, then the addition of workers of different types might begin at different times and, as a result, we allow for this generalization in the model.

Since $w(x)$ is a step function with possible jumps only at discrete time points, we can rewrite Equation (13) as

$$H(t)=\begin{cases}0 & \text{if } t<T_s\\ \sum_{j=T_s}^{[t]-1} w(j)+w([t])(t-[t]) & \text{if } T_s\le t\le T_E.\end{cases} \tag{14}$$

If there are $n(x)$ nonproductive workers at time x, then $n(x)\,dx$ is the amount of nonproductive effort expended by workers going thru the learning phase in the interval $(x, x+dx)$. Letting $H_{NS}(t)$ denote the total nonproductive effort expended by nonproductive workers by time t, we have

$$H_{NS}(t)=\begin{cases}0 & \text{if } t<T_s\\ \int_{T_s}^{t} n(x)\,dx & \text{if } T_s\le t\le T_E.\end{cases} \tag{15}$$

Since $n(x)$ is a step function with possible jumps only at time points of the form j, $j+L-[L]$, $j+1$, $j+1+L-[L]$, . . . , etc., where $j=T_s, T_s+1, \ldots$, we can rewrite Equation (15), if $T_s\le t\le T_E$, as

$$H_{NS}(t)=\sum_{j=T_s}^{[t]-1}\left((L-[L])n(j)+(1-L+[L])n(j+L-[L])\right)$$

$$+\begin{cases}n([t])(t-[t]) & \text{if } [t]\le t<[t]+L-[L]\\ n([t])(L-[L])+(t-[t]-L+[L])n([t]+L-[L]) & \text{if } [t]+L-[L]\le t<[t]+1.\end{cases} \tag{16}$$

VI. EFFICIENCY OF A PRODUCTIVE WORKER

Once a worker is trained (i.e., productive in the sense of having spent at least L time units on the program) it can be expected that he will have some interaction with those workers of the same type who are not yet trained and may even participate in the conduct of their training. For this reason it seems plausible to introduce a degradation factor, or what we shall choose to call an efficiency factor, to account for a partial loss in productivity of the trained worker when the ratio of nontrained to pro-

ductive workers of the same type exceeds a specified threshold. For example, consider a fully productive design engineer working with a group of four other design engineers. If all four of these design engineers are already trained, then this particular engineer ought to be able to perform at 100-percent efficiency; however, if all four of these engineers are not yet trained, then we can expect the trained engineer to operate at an efficiency level considerably less than 100 percent.

Consequently, let us arbitrarily suppose that for a given threshold K (called the buildup threshold) dependent on the type of worker, a productive worker has an efficiency factor, $e(t)$ say, equal to 1 if $n(t)/p(t) \leq K$, and has an efficiency factor, $e(t)$, proportional to the fraction of workers of the same type which are productive if $n(t)/p(t) > K$. More precisely, we define

$$e(t) = \begin{cases} 1 & \text{if } n(t)/p(t) \leq K \\ \alpha p(t)/n(t) & \text{if } n(t)/p(t) > K, \end{cases} \tag{17}$$

where α is the constant of proportionality. Because of the desire that $e(t)$ be a continuous function, we must have $e(t) \to 1$ as $n(t)/p(t) \to +K$. This implies that $\alpha = K + 1$ since taking the limit as $n(t)/p(t) \to +K$, we obtain

$$1 = \lim e(t) = \lim \alpha/(n(t)/p(t) + 1) = \frac{\alpha}{K+1}. \tag{18}$$

Therefore, the efficiency factor is defined as

$$e(t) = \begin{cases} 1 & \text{if } n(t)/p(t) \leq K \\ (K+1)p(t)/n(t) & \text{if } n(t)/p(t) > K. \end{cases} \tag{19}$$

It is important to observe that if $n(t)/p(t) > K$, then $(K+1)p(t)/n(t) < 1$. Furthermore, it is easily shown that $e(t)$ has the form described in Figure 4.

The choice of form of $e(t)$ as defined in (19) is somewhat arbitrary and, of course, there are many other functions which approach 1 as $n(t)/p(t) \to +K$. For example, two such functions are $\exp -(n(t)/p(t) - K)$ and $1/(1 - K + n(t)/p(t))$. No empirical evidence is known to this author to suggest which function would be the best to use in defining $e(t)$ and consequently the choice made is based upon intuitive judgment.

The concept of worker efficiency has been considered many times elsewhere and, in particular, in References 2–4. Jewett [3] assumes a constant efficiency factor for experienced, new, overtime, and subcontracted workers, but these factors are independent of the time period. In contrast to this, Haehling von Lanzenauer [2] assumes constant worker productivity for each time period, and Lundgren and Schneider [4] consider the situation where a new hire has efficiency e_i in the ith time period.

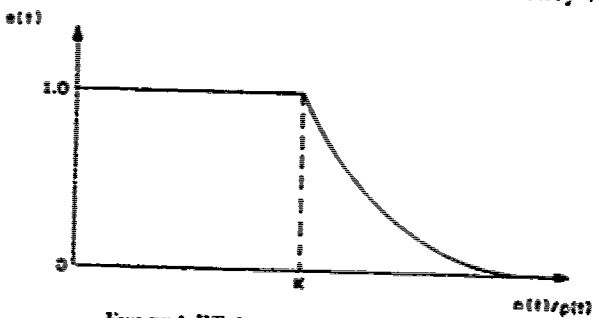


FIGURE 4. Efficiency of a productive worker

where $e_1 \leq e_2 \leq \ldots \leq e_n$. The latter state as their rationale that "since efficiency of the new employee is based on a learning curve application, the overall time span is divided into appropriate periods to accomodate his increasing productivity." None of these papers consider the potential effect on the efficiency of trained workers due to the addition of new workers to the organization, and in Ref. [4] all new workers are hired initially rather than gradually according to some buildup plan.

In summary, in this paper we are assuming that

(1) new workers have efficiency = 0 during their learning period and efficiency = 1.0 after their learning period is completed; and

(2) given that we define productive workers to be those who have completed their learning period, then productive workers have time dependent efficiency given by $e(t)$ in Equation (19).

Finally, the choice of the buildup threshold can be expected to be influenced by many factors of which some typical ones can be expected to be:

(1) overstaffing
(2) the classification mix of personnel
(3) the rate of infusion of new workers
(4) inadequate work planning and milestone identification
(5) inadequate problem definition
(6) the intensity of activity to meet program objectives
(7) the match between personnel training and work assignments
(8) availability of technical equipment and facilities
(9) unavailability of adequate work standards
(10) program phase (concept formulation, definition, acquisition)
(11) frequency and type of technical changes to the scope of effort
(12) the time spent in communication with coworkers
(13) the time spent in new orientation meetings.

VII. DETERMINATION OF TOTAL NONPRODUCTIVE LABOR UNIT EXPENDITURES

Because of the assumption of degraded efficiency of productive workers when the nonproductive workers "outnumber" in the above defined sense the productive workers, it is possible to obtain nonproductive effort from productive workers. This is the penalty one would pay for building up too rapidly so as to cause the ratio $n(t)/p(t)$ to exceed the buildup threshold. Thus, if there are $p(x)$ productive workers at time x, then in the interval $(x, x+dx)$ they will expend $p(x)(1-e(x))dx$ labor units for nonproductive effort and so the total amount $H_{NP}(t)$, say, of nonproductive effort by productive workers by time t can be expressed as

$$H_{NP}(t) = \begin{cases} 0 & \text{if } t < T_S \\ \int_{T_S}^{t} p(x)(1-e(x))dx & \text{if } T_S \leq t \leq T_E. \end{cases} \tag{20}$$

Since $p(x)$, like $n(x)$, is a step function with possible jumps only at time points of the form $j, j+L-[L], j+1, j+1+L-[L], \ldots$, etc., where $j = T_S, T_S+1, \ldots$, we can rewrite Equation (20) if $T_S \leq t \leq T_E$ as

$$\begin{aligned} H_{NP}(t) = &\sum_{j=T_S}^{[t]-1} \left((L-[L])p(j)(1-e(j)) + (1-L+[L])p(j+L-[L])(1-e(j+L-[L]))\right) \\ &+ \begin{cases} p([t])(t-[t])(1-e([t])) & \text{if } [t] \leq t < [t]+L-[L] \\ p([t])(L-[L])(1-e([t])) + (t-[t]-L+[L])p([t]+L-[L])(1-e([t]+L-[L])) & \text{if } [t]+L-[L] \leq t < [t]+1. \end{cases} \end{aligned} \tag{21}$$

Therefore, the total amount of nonproductive effort by time t, denoted by $H_N(t)$, is given by

$$H_N(t) = H_{NA}(t) + H_{NP}(t) \tag{22}$$

Similarly, the total amount of productive effort by time t is

$$H_P(t) = \begin{cases} 0 & \text{if } t < T_s \\ \int_{T_s}^{t} p(x)e(x)\,dx & \text{if } T_s \leq t \leq T_L \end{cases} \tag{23}$$

or, equivalently, by

$$H_P(t) = H(t) - H_N(t). \tag{24}$$

The ratio $H_N(t)/H(t)$, for $T_s \leq t \leq T_L$, provides a useful measure of the fraction of the total labor units expended for nonproductive effort by time t. If R denotes the average worker labor rate (in dollars per labor unit), then $H_N(t)R$ is an estimate of the total labor dollars spent for nonproductive effort by time t. Since workers must be trained, we can always expect to pay $H_{NA}(t)R$ for training, but by careful design of the manpower plan it may be possible to not incur the penalty cost $H_{NP}(t)R$ (i.e., by choosing U appropriately, we can insure that $u(t)/p(t)$ never exceeds K and so $H_{NP}(t) = 0$ for all t).

VIII. DETERMINATION OF OPTIMAL MANPOWER PLANS

There are, no doubt, many criteria relative to which one can determine optimal manpower plans such as the plan of least cost, the one for which the buildup threshold is never exceeded, the one with the least amount of nonproductive effort over a given time period, etc. For example, Jewett [3] considers the following problem: Given a single set of time-wise manpower requirements, a workload for a contract during a specified time period, what is the manpower schedule which minimizes the total cost of performing exactly this workload? Also, Lundgren and Schneider [4] consider the problem: Giving consideration to alternatives such as hiring new people, working overtime, or doing both, find the policy that minimizes the total cost subject to fixed output or demand at the end of a given time period.

In particular, suppose one is interested in finding the least cost manpower plan over the interval $[0, T]$ (i.e., the plan for which $H(T)$ is a minimum) that meets a specified set of schedule milestones. Let the time of the jth schedule milestone be denoted by t_j and suppose that an amount of productive effort, denoted by $C(t_j)$ must be expended by time t_j in order to meet this milestone. If we are given $p(0)$, the time T_P of earliest possible peak, N milestones with productive effort objectives $C(t_1), \ldots, C(t_N)$, and assume that $u(t)$ cannot decrease prior to T_P and cannot increase after T_P, then we can formulate the problem as follows:

$$\text{minimize } H(T) \tag{25}$$

subject to

$$u(0) - p(0) \geq 0 \tag{26}$$

$$u(t) - u(t-1) \geq 0 \quad \text{for } t = 1, 2, \ldots, T_P \tag{27}$$

$$u(t) - u(t-1) \leq 0 \quad \text{for } t = T_P + 1, T_P + 2, \ldots, T \tag{28}$$

$$H_P(t_j) = C(t_j) \quad \text{for } j = 1, 2, \ldots, N \tag{29}$$

$$u(t) \geq 0 \quad \text{for each integer } t. \tag{30}$$

This is a typical nonlinear programming problem in the integer variables $u(0), u(1), \ldots, u(T)$.

However, if we specify that the learning time L is integral and that the buildup threshold must not be exceeded during the buildup process, then Equation (29) becomes linear in the $w(t)$'s. This is facilitated by adding the constraint that $n(t)/p(t) \leq K$ for $t=0, 1, 2, \ldots, T$ or, equivalently,

$$(1+K)p(t)-w(t) \geq 0 \qquad \text{for } t=0, 1, 2, \ldots, T. \tag{31}$$

This type of constraint has been considered before by Purkiss [5] when he imposed the condition that the number of trainees cannot exceed some fixed proportion K of the trained men.

Before showing that the problem defined by (25)–(31) is an integer linear programming problem (see Ref. [1]), observe that an alternative, and perhaps more realistic, form of Equation (29) is for given constants Δ_j to write

$$C(t_j)-\Delta_j \leq H_P(t_j) \leq C(t_j)+\Delta_j \tag{32}$$

for $j=1, 2, \ldots, N$, i.e., the total productive effort by time t_j should be within Δ_j of the desired goal $C(t_j)$.

For integral T and $T_s=0$ the objective function is linear, because we have from Equation (14) that

$$H(T)=\sum_{j=0}^{T-1} w(j). \tag{33}$$

Since we don't allow the buildup threshold to be exceeded, it follows that $H_{wp}(t)=0$ and so from Equations (16), (22), (24), and (31) we obtain

$$H_P(t)=H(t)-H_{ws}(t) \tag{34}$$

$$=\sum_{j=0}^{t-1} w(j)-\sum_{j=0}^{t-L} w(j)=\sum_{j=t-L}^{t-1} p(j). \tag{35}$$

since L is assumed to be an integer. Consequently, it suffices to show that $p(t)$ is linear in the $w(t)$'s. Using the definitions of $U(j)$ and $D(j)$ along with (27) and (28), it is easily shown from Equation (5) that, if $L \leq T_P$, then

$$p(t)=\begin{cases} p(0) & \text{if } 1 \leq t < L \\ w(t-L) & \text{if } L \leq t \leq T_P \\ w(t-L)+w(t)-w(T_P) & \text{if } T_P < t \leq T_P+L \\ w(t) & \text{if } T_P+L < t. \end{cases} \tag{36}$$

and, if $T_P < L$, then

$$p(t)=\begin{cases} p(0) & \text{if } 1 \leq t < T_P \\ p(0)+w(t)-w(T_P) & \text{if } T_P \leq t < L \\ w(t-L)+w(t)-w(T_P) & \text{if } L \leq t < T_P+L \\ w(t) & \text{if } T_P+L \leq t. \end{cases} \tag{37}$$

Therefore, from Equations (33) and (35)–(37) it follows that (25) and (29) are linear functions of $w(t)$ and so (25)–(31) defines an integer LP problem. The solution can be obtained by using Gomory's Method (see Ref. [1]).

IX. DETERMINATION OF THE OPTIMUM BUILDUP RATE

In discussing optimal manpower plans consideration is sometimes given to the concept of an optimum buildup rate. Two possible criteria for such a rate are: (1) the fraction of the total labor force

which is trained (i.e., productive) is never less than a given level α, that is, $p(t)/w(t) \geq \alpha$ for all $t \geq T_S$, and (2) the fraction of labor expended for productive effort is never less than a given level β, that is, $H_P(t)/H(t) \geq \beta$ for all $t \geq T_S$. Probably, the most important criterion, since it is certainly not desirable to obtain nonproductive effort from productive people, is that the ratio $n(t)/p(t)$ never exceed the buildup threshold. This means that the only nonproductive effort obtained is due to the training of new workers and not due to an excessive buildup rate which would cause productive people to become partially nonproductive.

Thus, as an illustration, let us consider the problem of determining the maximum buildup rate, U^* say, such that $n(t)/p(t) \leq K$ for all $t \geq T_S$. For convenience, we will assume $T_S = 0$ and initially that $n(0)/p(0) \leq K$. If the organization builds up at the rate U^*, then the maximum size of the organization will be $w(T_P) = w(0) + U^*(T_P - T_I + 1)$, where T_I is the time at which the buildup is begun. One must consider the following four cases to determine the optimal buildup rate, in which we restrict ourselves to considering only $t \leq T_P$ since the buildup ends at T_P:

CASE 1: $L \leq T_I$ and $T_I + L \leq T_P$.

(a) If $0 \leq t < L$, then $n(t)/p(t) = n(0)/p(0)$, since $t < T_I$.

(b) If $L \leq t < T_I$, then $n(t)/p(t) = 0$, since the initial number of untrained workers have become trained and the buildup has not yet begun.

(c) If $T_I \leq t < T_I + L$, then $n(t)/p(t) = U[t - T_I + 1]/w(0)$ and so

$$\max\left\{\frac{n(t)}{p(t)} : T_I \leq t < T_I + L\right\} = \begin{cases} UL & \text{if } L \text{ is integral} \\ U[L+1] & \text{if } L \text{ is nonintegral.} \end{cases} \tag{38}$$

Therefore, in order that $n(t)/p(t) \leq K$ for $T_I \leq t < T_I + L$, it suffices to choose

$$U = \begin{cases} Kw(0)/L & \text{if } L \text{ is integral} \\ Kw(0)/[L+1] & \text{if } L \text{ is nonintegral.} \end{cases} \tag{39}$$

(d) If $T_I + L \leq t \leq T_P$, then

$$\frac{n(t)}{p(t)} = \frac{U[t - T_I + 1] - U[t - T_I + 1 - L]}{w(0) + U[t - T_I + 1 - L]} = \frac{U([t] - [t-L])}{w(0) + U[t - T_I + 1 - L]}. \tag{40}$$

Since

$$0 \leq [t] - [t-L] \leq \begin{cases} L & \text{if } L \text{ is integral} \\ [L+1] & \text{if } L \text{ is nonintegral.} \end{cases} \tag{41}$$

it follows that

$$\max\left\{\frac{n(t)}{p(t)} : T_I + L \leq t \leq T_P\right\} \leq \max\left\{\frac{n(t)}{p(t)} : T_I \leq t < T_I + L\right\}. \tag{42}$$

Therefore, in order that $n(t)/p(t)$ doesn't exceed K, it suffices to choose U according to (39). In summary, we have

$$U^* = \begin{cases} Kw(0)/L & \text{if } L \text{ is integral} \\ Kw(0)/[L+1] & \text{if } L \text{ is nonintegral.} \end{cases} \tag{43}$$

CASE 2: $L \leq T_I$ and $T_I + L > T_P$.

(a) If $0 \leq t < L$, then $n(t)/p(t) = n(0)/p(0)$, since $t < T_I$.

(b) If $L \leq t < T_I$, then $n(t)/p(t) = 0$, since the initial number of untrained workers are now trained and the buildup hasn't started.

(c) If $T_{t'} \leq t \leq T_P$, then $n(t)/p(t) = U[t - T_{t'} + 1]/w(o)$ and so

$$\max\left\{\frac{n(t)}{p(t)} : T_{t'} \leq t \leq T_P\right\} = \frac{U(T_P - T_{t'} + 1)}{w(o)}. \tag{44}$$

Therefore, in order that $n(t)/p(t) \leq K$ for $T_{t'} \leq t \leq T_P$, it suffices to choose

$$U = Kw(o)/(T_P - T_{t'} + 1). \tag{45}$$

In summary, we choose $U^* = U$ as given by (45).

CASE 3: $L > T_{t'}$ and $T_{t'} + L \leq T_P$.

(a) If $0 \leq t < T_{t'}$, then $n(t)/p(t) = n(o)/p(o)$.

(b) If $T_{t'} \leq t < L$, then

$$\frac{n(t)}{p(t)} = \frac{n(o) + U[t - T_{t'} + 1]}{p(o)} \tag{46}$$

and so

$$\max\left\{\frac{n(t)}{p(t)} : T_{t'} \leq t < L\right\} = \begin{cases} \dfrac{n(o) + U(L - T_{t'})}{p(o)} & \text{if } L \text{ is integral} \\ \dfrac{n(o) + U([L] - T_{t'} + 1)}{p(o)} & \text{if } L \text{ is nonintegral.} \end{cases} \tag{47}$$

Therefore, it suffices to choose

$$U = \begin{cases} \dfrac{Kp(o) - n(o)}{L - T_{t'}} & \text{if } L \text{ is integral} \\ \dfrac{Kp(o) - n(o)}{[L] - T_{t'} + 1} & \text{if } L \text{ is nonintegral.} \end{cases} \tag{48}$$

(c) If $L \leq t < T_{t'} + L$, then $n(t)/p(t) = U[t - T_{t'} + 1]/w(o)$ and so

$$\max\left\{\frac{n(t)}{p(t)} : L \leq t < T_{t'} + L\right\} = \begin{cases} UL/w(o) & \text{if } L \text{ is integral} \\ U[L+1]/w(o) & \text{if } L \text{ is nonintegral.} \end{cases} \tag{49}$$

Therefore, it suffices to choose

$$U = \begin{cases} Kw(o)/L & \text{if } L \text{ is integral} \\ Kw(o)/[L+1] & \text{if } L \text{ is nonintegral} \end{cases} \tag{50}$$

(d) If $T_{t'} + L \leq t \leq T_P$, then

$$\frac{n(t)}{p(t)} = \frac{U([t] - [t - L])}{w(o) + U[t - T_{t'} + 1 - L]}. \tag{51}$$

From Equations (41) and (42) it follows that $n(t)/p(t)$ won't exceed K if we choose U according to (50).

In summary, we choose

$$U^* = \begin{cases} \min\left\{ \dfrac{Kp(o) - n(o)}{L - T_t} \cdot \dfrac{Kn(o)}{L} \right\} & \text{if } L \text{ is integral} \\ \min\left\{ \dfrac{Kp(o) - n(o)}{[L] - T_t + 1} \cdot \dfrac{Kn(o)}{[L+1]} \right\} & \text{if } L \text{ is nonintegral.} \end{cases} \tag{52}$$

CASE 4: $L > T_t$ and $T_t + L > T_p$.

(a) If $0 \leq t < T_t$ and $n(t)/p(t) = n(o)/p(o)$.

(b) If $T_t \leq t < L$, then $n(t)/p(t)$ is given by (46) and so we choose U according to Equation (48).

(c) If $L \leq t \leq T_p$, then $n(t)/p(t) = U[t - T_t + 1]/n(o)$, and so, referring to Case 2(c), we choose U according to Equation (45).

In summary, we choose

$$U^* = \begin{cases} \min\left\{ \dfrac{Kn(o)}{T_p - T_t + 1} \cdot \dfrac{Kp(o) - n(o)}{L - T_t} \right\} & \text{if } L \text{ is integral} \\ \min\left\{ \dfrac{Kn(o)}{T_p - T_t + 1} \cdot \dfrac{Kp(o) - n(o)}{[L] - T_t + 1} \right\} & \text{if } L \text{ is nonintegral.} \end{cases} \tag{53}$$

X. USES OF THE MODEL

The model, as structured, yields five basic output quantities, namely: (1) the total number of nonproductive workers; (2) the number of productive workers; (3) the cumulative labor unit expenditure; (4) the cumulative expenditure of labor units for training (i.e., nonproductive effort by nonproductive workers); (5) the cumulative expenditure of nonproductive labor units (i.e., nonproductive effort by nonproductive and productive workers). From these quantities can be computed such interesting quantities as the fraction of the total labor force which is trained (i.e., productive), and the fraction of the cumulative labor units (or labor cost) expended for nonproductive effort. Parametric analyses can be performed on the latter quantities to determine the combinations of input variables which optimize these in some defined sense. If the optimum buildup rate is defined to be the maximum value of the buildup rate, say U_{max}, then the model can be used to determine the maximum feasible organization size; namely, in the case of one worker type, $n(T_s) + U_{max}(T_p - T_t + 1)$.

Another use of the model would be in the performance of hindsight analyses on past programs. In particular, given the labor cost history of the program (i.e., the cumulative direct labor dollar expenditure as a function of time) and estimates of the model input variables, one can then estimate the total labor dollars spent for nonproductive effort by time t using the estimate $C(t)H_s(t)/H(t)$, where $C(t)$ denotes the cumulative actual direct labor cost by time t. Another such estimator would be $RH_s(t)$ where R denotes the average direct labor rate for the time period $[0, t]$.

If a company has a specified set of milestones relative to a contract schedule which must be met, the model provides a tool for determining whether or not these can be achieved for a given manpower plan. For example, suppose 16 months after go-ahead is a critical milestone in the development of a particular type of electronic device. Past experience on similar types of devices has shown that approximately 34,500 man-months of productive effort are required to meet such a milestone. The program manager and his staff have worked out a manpower plan which calls for 1,000 initial workers, all of

which are productive (i.e., already trained), and a buildup rate of 300 workers per month, starting at the end of the first month (i.e., $T_0 = 1$), and continuing for 20 months. The learning time is estimated to be 2 months on the average and the buildup threshold is estimated to be 0.25. Using Equation (44), when $T_0 = 1$ and the buildup rate is U, we obtain for integral t

$$H(t) = \sum_{j=0}^{t} w(j) = w(o) + \sum_{j=1}^{t} w(j) \tag{54}$$

$$= w(o) + \sum_{j=1}^{t} (w(o) + jU) \tag{55}$$

$$= tw(o) + \frac{Ut(t+1)}{2}. \tag{56}$$

In particular, for $t = 16$, $w(o) = 1{,}000$ and $U = 300$, we obtain $H(16) = 42{,}400$ man-months of effort. From Table 1, we see that the fraction of effort productive by time 16 is 0.819 and so $H_p(16) = 34{,}726$ man-months. Therefore, the milestone can easily be met, but at the cost of $42{,}400 - 34{,}726 = 7{,}674$ additional man-months of effort, that is, to get 34,726 man-months of productive effort, the company must expend 42,400 man-months.

From Table 1, we observe that the milestone could not have been met within 15 months after go-ahead since (0.810) (42,400) ... 34,344 man-months. On the other hand, if the program manager and his staff were conservative in their estimate of the buildup threshold K, and the true threshold is approximately 1.0, then we observe that the milestone could be met within 14 months after go-ahead since (0.818) (42,400) = 34,683 man-months.

TABLE 1. Fraction Of Effort Productive When $w(o) = p(o) = 1000$, $L = 2.0$ And $U = 300$

Time From Start (t)	Buildup Threshold			
	0.25	0.50	0.75	1.0
1	1.000	1.000	1.000	1.000
2	0.853	0.870	0.870	0.870
3	0.703	0.753	0.769	0.769
4	0.665	0.731	0.741	0.741
5	0.664	0.730	0.738	0.738
6	0.678	0.737	0.743	0.743
7	0.697	0.747	0.752	0.752
8	0.718	0.758	0.762	0.762
9	0.736	0.770	0.773	0.773
10	0.752	0.780	0.783	0.783
11	0.766	0.790	0.793	0.793
12	0.779	0.800	0.802	0.802
13	0.791	0.809	0.810	0.810
14	0.801	0.817	0.818	0.818
15	0.810	0.824	0.826	0.826
16	0.819	0.831	0.833	0.833

There are advantages to initially starting the program with a trained or productive group of workers, however, because of the sometimes nonzero time lapses between the end of one program and the start

of a subsequent one, a company is frequently faced with the problem of supporting a basic labor force during such an interim period while no contractual funds may be available. One rationale for determining what the size of this basic labor force should be can be developed as follows. Let $M(x)$ denote the total estimated cost for nonproductive effort (i.e., $H_S(T_P+L)$ on a proposed program if the initial number of productive workers is x (i.e., $p(T_S)=x$). Then $M(x)-M(x+\Delta)$ denotes the savings in expenditure for nonproductive effort if the program begins with $x+\Delta$ rather than x productive workers. If time is expressed in months and $M(x)$ and $M(x+\Delta)$ in man-months, then this savings is equivalent to supporting the additional Δ workers for $(M(x)-M(x+\Delta))/\Delta$ months prior to program go-ahead, that is, it is to the company's advantage to start the program with $x+\Delta$ productive workers rather than x if the time required to support these additional productive workers does not exceed $(M(x)-M(x+\Delta))/\Delta$.

To illustrate this application of the model, suppose a company plans to buildup a program organization starting with 500 workers, all initially productive, to a maximum size of 10,000 workers 24 months after go-ahead, i.e., setting $T_S=0$ and $T_P=1$, $p(o)=w(o)=500$, $T_P=24$ and $w(T_P)=10{,}000$. This implies a buildup rate of approximately 396 $\left(=\dfrac{w(T_P)-p(0)}{24}\right)$ workers per month. Suppose the learning time is estimated to be 1.5 months and the buildup threshold is 0.1. The organization planners are interested in determining whether or not it would be more advantageous to start with a larger number of initially productive workers and, if so how much time would be required prior to go-ahead to support this increase in the initial workers productive. They are interested in considering the range $500 \leq p(0) \leq 5000$. Using Equation (22) with $t=T_P+L$, the total nonproductive effort was computed for $p(o)$ equal to integral multiples of 500 in this specified range and is given in Table 2 as $M(p(0))$. Setting $x=500$ and $\Delta=500, 1000, \ldots, 4500$, in Table 3 are presented the corresponding values of $M(x)-M(x+\Delta)$ and the ratio $(M(x)-M(x+\Delta))/\Delta$. From this table we see that 1,500 additional productive workers can be supported prior to go-ahead for the largest possible time (a little over 2.5 months) implying that, if there can be expected to be a 2–3 month delay before the start of the program, then to start the program with 2,000 productive workers (hence a buildup rate of 333 heads per month) would be more efficient than starting with only 500.

TABLE 2. Total Program Nonproductive Effort (man-months)

$p(o)$	$M(p(o))$
500	17,606
1,000	16,333
1,500	14,990
2,000	13,620
2,500	12,363
3,000	11,172
3,500	10,086
4,000	9,112
4,500	8,285
5,000	7,488

TABLE 3. Savings in Nonproductive Effort and Maximum Support Time ($x=500$)

Δ	$M(x)-M(x+\Delta)$	$(M(x)-M(x+\Delta))/\Delta$
500	1.273	2.546
1,000	2.616	2.616
1,500	3.986	2.657
2,000	5.243	2.622
2,500	6.434	2.574
3,000	7.524	2.508
3,500	8.494	2.427
4,000	9.321	2.330
4,500	10.118	2.248

We have presented a few applications of the model developed here to suggest how it can be used and perhaps these can stimulate the interested reader to find further uses and applications.

REFERENCES

[1] Dantzig, G. B., *Linear Programming and Extensions* (Princeton University Press, Princeton, N.J., 1963).

[2] Haehling von Lanzenauer, C., "Production and Employment Scheduling in Multistage Production Systems," Nav. Res. Log. Quart., **17**, 193–198 (June 1970).

[3] Jewett, R. F., "A Minimum Risk Manpower Scheduling Technique," Management Science, **13**, 578–592 (June 1967).

[4] Lundgren, E. F. and J. V. Schneider, "A Marginal Cost Model for the Hiring-Overtime Decision," Management Science, **17**, 399–405 (Feb. 1971).

[5] Purkiss, C. J., "Approaches to Recruitment, Training, and Redeployment Planning in an Industry" in *Manpower Research in a Defence Context*, Edited by Dr. N. A. B. Wilson (American Elsevier Publishing Company, Inc., New York, 1969).

[6] Rau, J. G., "Mathematical Modeling of the Program Buildup Process," paper presented at the ORSA/TIMS/LACC Joint Meeting in Los Angeles, California on 4 Oct. 1968.

ON MODELS FOR BUSINESS FAILURE DATA*

Satya D. Dubey

Department of Industrial Engineering and Operations Research
New York University

ABSTRACT

It is pointed out in this paper that Lomax's hyperbolic function is a special case of both Compound Gamma and Compound Weibull distributions, and both of these distributions provide better models for Lomax's business failure data than his hyperbolic and exponential functions. Since his exponential function fails to yield a valid distribution function, a necessary condition is established to remedy this drawback. In the light of this result, his exponential function is modified in several ways. It is further shown that a natural complement of Lomax's exponential function does not suffer from this drawback.

1. INTRODUCTION

While analyzing data on failures of four types of business, Lomax proposed two functions which gave good fit to his data in [9]. His hyperbolic function

$$z(t)=\frac{b}{t+a},\ t\geq 0, \tag{1.1}$$

where a and b are positive constants, was found more appropriate for the data relating to retail, craft, and service groups while his exponential function

$$z(t)=ae^{-bt},\ t\geq 0, \tag{1.2}$$

where a and b are positive constants, gave better fit to the data on manufacturing trades. In this paper it will be pointed out that Lomax's hyperbolic function is a special case of both Compound Gamma and Compound Weibull distributions, and both of these distributions provide better models for Lomax's business failure data than his hyperbolic and exponential functions. In addition, several important points relating to Lomax's paper [9] will also be considered.

2. LOMAX'S HYPERBOLIC FUNCTION

From Reference [2] we obtain the probability density function (p.d.f.) of the Compound Gamma distribution as

$$f_T(t)=\begin{cases}\dfrac{a^b t^{a-1}}{B(a,b)[a+t]^{a+b}}, & t\geq 0,\ (a,a,b>0),\\ 0 & \text{elsewhere.}\end{cases} \tag{2.1}$$

*This research was partially supported by the Aerospace Research Laboratory, Office of Aerospace Research, U.S.A.F., under contract F33615-71-C-1174.

which reduces to the p.d.f. of the Lomax distribution corresponding to $z(t)=b/(t+a)$ for $\alpha=1$. This Lomax distribution enjoys the following property.

PROPERTY: Let $X_1, X_2, \ldots, X_n$ be n independent identically distributed random variables from the Lomax distribution with the parameters a and b, and let $Y_n=\min(X_1, X_2, \ldots, X_n)$. Then Y_n obeys the Lomax distribution with the parameters a and nb. Conversely, if Y_n has the Lomax distribution with the parameters α and β then each $X_i (i=1, 2, \ldots, n)$ obeys the Lomax distribution with the parameters α and βn^{-1}.

This property is very useful in life testing situations where failure data are generated by destructive testing requiring unduly long waiting period. The proof and the application of this type of result are discussed in Reference [4].

The above Lomax distribution may be considered to be Exponential-Gamma distribution. Later on, it will be shown that a Gamma-Exponential distribution is a better model for Lomax's data than his Exponential-Gamma distribution. It may be noted that both of these distributions are special cases of the above Compound Gamma distribution.

From Reference [3] we obtain the p.d.f. of the Compound Weibull distribution as

$$f_T(t)=\begin{cases}\dfrac{b\gamma a^b t^{\gamma-1}}{[a+t^\gamma]^{b+1}}, & t\geq 0,\ (a, b, \gamma>0),\\ 0 \text{ elsewhere,}\end{cases} \tag{2.2}$$

which reduces to the p.d.f. of the Lomax distribution for $\gamma=1$. Subsequently, it will be shown that the Weibull-Exponential distribution, which is a special case of the above Weibull-Gamma distribution and different from the above Lomax distribution, is a better model for Lomax's data than his hyperbolic function.

3. LOMAX'S EXPONENTIAL FUNCTION

Earlier, we have stated that Lomax's exponential function does not yield a valid distribution function. It is because it yields

$$F(\infty)=1-e^{-\frac{a}{b}}<1,$$

which is contrary to the usual assumption, $F(\infty)=1$, noting that $F(x)$ denotes the cumulative distribution function. For a given intensity function $z(t)$, we write

$$\int_{-\infty}^{x} z(t)dt=\int_{-\infty}^{x}\frac{f(t)dt}{1-F(t)},$$

which gives

$$F(x)=1-\exp\left[-\int_{-\infty}^{x} z(t)dt\right]. \tag{3.1}$$

Since $z(t)\geq 0$ for all t, $F(\infty)=1$ requires

$$\int_{-\infty}^{x} z(t)dt\to\infty \text{ as } x\to\infty, \tag{3.2}$$

which is a necessary condition for the existence of a valid distribution function. It is easy to see that Lomax's exponential function does not satisfy this condition. We note that $z(t)$, the conditional density function of failure probability with time, is also known as the intensity function in [7], and the *hazard*, or the *age-specific failure rate* function in [1].

The p.d.f. of a *modified* Lomax distribution is given by

$$f_T(t) = \begin{cases} \dfrac{ae^{-bt+\frac{a}{b}(e^{-bt}-1)}}{1-e^{-\frac{a}{b}}}, & t \geq 0, \\ 0 \text{ elsewhere.} \end{cases} \tag{3.3}$$

which yields $F(\infty) = 1$.

The modified Lomax distribution is now made more versatile by considering one of its parameters to be a random variable and using this fact to generate a compound modified Lomax distribution. Besides, whenever experimental data, applicable to p.d.f. (3.3), are suspected to have been influenced by some uncontrollable factors, its compound p.d.f. derived below, may provide an improved fit to such data. Thus we shall write the conditional p.d.f. of the modified Lomax distribution as

$$f_T(t|a) = \begin{cases} ae^{-bt+\frac{a}{b}(e^{-bt}-1)}\ (1-e^{-\frac{a}{b}})^{-1}, & t \geq 0, \\ 0 \text{ elsewhere.} \end{cases} \tag{3.4}$$

and the p.d.f. of a as

$$g(a) = \begin{cases} \dfrac{\beta^\alpha a^{\alpha-1}e^{-\beta a}}{\Gamma(\alpha)}, & a > 0, \ (\alpha, \beta > 0), \\ 0 \text{ elsewhere.} \end{cases} \tag{3.5}$$

which corresponds to the gamma probability distribution. Now the p.d.f. of the compound modified Lomax distribution is given by

$$\begin{aligned} f_T(t) &= \int_0^\infty f_T(t|a)g(a)da \\ &= \begin{cases} \displaystyle\sum_{j=0}^{\infty} \frac{\alpha\beta^\alpha b^{\alpha-1}e^{-bt}}{(\beta b+j+1-e^{-bt})^{\alpha+1}}, & t \geq 0, \\ 0 \text{ elsewhere.} \end{cases} \end{aligned} \tag{3.6}$$

Another way of modifying the Lomax intensity function is to choose $z(t) = c + ae^{-bt}$, where c is a positive constant. This form of $z(t)$ enjoys a useful characterization (see Reference [4]). One may also choose $z(t)$ differently to remedy this difficulty.

4. SOME NUMERICAL RESULTS

Lomax reported correlation coefficients between theoretical and observed values of linearized exponential and hyperbolic functions in [9]. In this paper it has been pointed out that Lomax's exponential function does not correspond to a valid distribution function. Consequently, several modifications of Lomax's exponential function have been proposed in section 3. One of these modifications

consists of adding a positive constant, c, to his exponential function. Now in Table 4.1 we present the correlation coefficients between observed and theoretical values of linearized functions $ln\ z(t) = \ln a - bt$ and $\ln (z(t) - c) = \ln a - bt$.

TABLE 4.1. Correlation Coefficients Corresponding to Exponential and Modified Exponential Functions

Type of Business	Exponential Function	A Modified Exponential Function	
	$z(t) = ae^{-bt}$	$z(t) = c + ae^{-bt}$	c
Retail	0.91	0.99	0.08
Manufacturing	0.96	0.96	0.001
Craft	0.93	0.95	0.08
Service	0.91	0.96	0.10

The values of c reported in Table 4.1 were chosen by carefully examining Lomax's data ([9], Table 3). One could determine the optimum values of c which would insure the maximum attainable correlation coefficients for these data. The author has decided against this because his other models easily yield higher values of correlation coefficients in all cases (see Table 4.2).

In case of the Gamma-Exponential distribution we use the transformation

$$\ln (F(t)/f(t)) = (t/\alpha) + (t^2/a\alpha), \tag{4.1}$$

which yields pseudo least squares estimators for a and α, explicitly. Lomax's data ([9], Tables 1 and 2) are used for this purpose. In case of the Weibull-Exponential distribution we use the transformation

$$\ln (F(t)/R(t)) = \gamma \ln t - \ln a, \tag{4.2}$$

which yields pseudo least squares estimators for γ and a, explicitly. Lomax's data ([9], Table 1) are used for this purpose. The expression $R(t) = 1 - F(t)$ is called the reliability function and the pseudo least squares estimators are the usual least squares estimators in terms of transformed data. The detailed work is reported in Reference [5]. Here we give the correlation coefficients between observed and theoretical values of Lomax's linearized exponential and hyperbolic functions and expressions (4.1) and (4.2).

TABLE 4.2. Correlation Coefficients Corresponding to Following Four Models

Type of Business	Lomax's Functions		Gamma-Exponential	Weibull-Exponential
	Exponential	Hyperbolic		
Retail	0.9067	0.9900	0.9988	0.9993
Manufacturing	0.9602	0.8242	0.9994	0.9983
Craft	0.9283	0.9911	0.9998	0.9998
Service	0.9124	0.9794	0.9991	0.9979

Entries of Table 4.2 clearly show that both Gamma-Exponential and Weibull-Exponential distributions describe Lomax's data better than his exponential and hyperbolic functions.

5. NATURAL COMPLEMENT OF LOMAX'S EXPONENTIAL FUNCTION

Whereas the Lomax exponential function did not yield a valid distribution function, the natural complement of his exponential function does yield a valid distribution function and it seems to be useful. We consider the natural complement of the Lomax exponential function to be

$$z(t) = ae^{bt},\ t \geq 0,\ (a, b > 0). \tag{5.1}$$

The above intensity function $z(t)$ generates the valid distribution function since

$$\int_0^x ae^{bt}dt \to \infty \text{ as } x \to \infty.$$

Corresponding to intensity function (5.1) we get the p.d.f. as

$$f_1(t) = \begin{cases} ae^{bt - \frac{a}{b}(e^{bt}-1)},\ t \geq 0, \\ 0 \text{ elsewhere.} \end{cases} \tag{5.2}$$

Let the parameter a of p.d.f. (5.2) obey the gamma p.d.f. of form (3.5). Then

$$f_2(t) = \begin{cases} \dfrac{\alpha\beta^{\alpha}b^{\alpha-1}e^{bt}}{(\beta b - 1 + e^{bt})^{\alpha+1}},\ t \geq 0, \\ 0 \text{ elsewhere.} \end{cases} \tag{5.3}$$

Its intensity function is given by

$$z(t) = \frac{d}{1 + he^{-bt}}. \tag{5.4}$$

where $d = ab$ and $h = b\beta - 1$. Expression (5.4) is known in the statistical literature as *logistic curve* (see Reference [8], pp. 658–661). Some other results pertaining to this topic are discussed in References [4] and [6].

REFERENCES

[1] Buckland, W. R., *Statistical Assessment of the Life Characteristic*, a bibliographic guide (Hafner Publishing Company, New York, 1964).

[2] Dubey, S. D., "Compound Gamma, Beta and F Distributions," Metrika, **16**, 27–31 (1970).

[3] Dubey, S. D., "A Compound Weibull Distribution," Nav. Res. Log. Quart., **15**, 179–188 (1968).

[4] Dubey, S. D., "Characterization Theorems for Several Distributions and Their Applications," J. Indust. Math. Soc., **16**, 1–22 (1966).

[5] Dubey, S. D., "Transformations for Estimation of Parameters," J. Indian Stat. Assoc., **4**, 109–124 (1966).

[6] Dubey, S. D., "A New Derivation of the Logistic Distribution," Nav. Res. Log. Quart., **16**, 37–40 (1969).

[7] Gumbel, E. J., *Statistics of Extremes* (Columbia University Press, New York, 1958).

[8] Hald, A., *Statistical Theory with Engineering Applications* (John Wiley & Sons, Inc., 1952).

[9] Lomax, K. S., "Business Failures: Another Example of the Analysis of Failure Data," J. Am. Stat. Assoc., **49**, 847–852 (1954).

A NOTE ON A COMPARISON OF CONFIDENCE INTERVAL TECHNIQUES IN TRUNCATED LIFE TESTS

W. E. Coleman,

T. Jayachandran,

and

D. R. Barr

Naval Postgraduate School
Monterey, California

ABSTRACT

Several approximate procedures are available in the literature for obtaining confidence intervals for the parameter λ of an exponential distribution based on time truncated samples. This paper contains the results of an empirical study comparing three of these procedures.

1. INTRODUCTION

In life testing applications, it is frequently desired to obtain a confidence interval for the parameter λ of an exponential distribution. In case a test plan is used for which all the observations are truncated at the same time point t_0, several approximate confidence interval procedures are available in the statistical literature. The purpose of this note is to report the results of an empirical study of the performances of three of these procedures with respect to the expected length of the interval, the variance of the interval length and the coverage probability.

The general setting of the problem is as follows: suppose the random variables $T_1, T_2, \ldots, T_n$ are independent and identically exponentially distributed with mean λ^{-1}. For $i = 1, 2, \ldots, n$, let X_i be equal to T_i truncated at t_0, and let Y_i be the Bernoulli random variable which is 1 if and only if $X_i < t_0$. We wish to find confidence intervals for λ, based on the X_i and Y_i.

In what follows, three confidence interval procedures are described, and some results of an empirical study of their performances are presented.

2. CONFIDENCE INTERVAL PROCEDURES

PROCEDURE 1: This procedure is obtained as a special case of a solution to a more general problem that was derived by Halperin [1]. The random variable $Y = \sum_{i=1}^{n} Y_i$ has a binomial distribution with parameters n and $p = 1 - e^{-\lambda t_0}$. Standard techniques can be used to obtain a $100(1-\alpha)$ percent confidence interval for p as

$$P[a(Y) < p < b(Y)] \geq 1 - \alpha.$$

Since $p = 1 - e^{-\lambda t_0}$, an inversion can be made which results in $\lambda_L = \dfrac{-\ln(1-a)}{t_0}$ and $\lambda_U = \dfrac{-\ln(1-b)}{t_0}$ as lower and upper $100(1-\alpha)$ percent confidence limits for λ.

PROCEDURE 2: Rubenstein [2] showed that

$$\hat{\lambda} = \frac{\Sigma Y_i}{\Sigma N_i}\left[1 + \frac{1}{2n}\right]^{-1}$$

is an approximately unbiased estimator of λ. He noted that for $\lambda t_0 \leq 1$, ΣY_i is nearly a Poisson random variable, so a confidence interval procedure for a Poisson parameter due to Wilks [3] was used to obtain the approximate confidence limits

$$\lambda_L = [2\hat{\lambda} + z^2/C - (4\hat{\lambda}z^2/C + z^4/C^2)^{1/2}]/2$$

$$\lambda_U = [2\hat{\lambda} + z^2/C + (4\hat{\lambda}z^2/C + z^4/C^2)^{1/2}]/2.$$

Where z is the $100(1-\alpha)$th percentage point of the standard normal distribution and $C = (\Sigma N_i)^{1/2}$.

PROCEDURE 3: We employ terminology commonly used in the literature of life testing in describing this procedure. Imagine that the random variables $X_1, X_2, \ldots, X_n$ are observed sequentially. That is, imagine that a randomly selected item is put on test and is replaced with a similar item at failure or after a period of t_0 has elapsed, whichever occurs first. If this process were continued, the arrival process of failures would be a Poisson process, so the time to k^{th} failure (for k fixed) would have a gamma distribution. (Testing to k^{th} failure in this situation could be described roughly as a combination of item censoring and time truncation.) Since we are assuming that exactly n items are to be tested, the experiment is stopped after a random amount of time, and the number K of observed failures is a random variable. It would appear, however, that, given $K = k$, the distribution of the time W'_k until k failures have arrived can be approximated by a gamma distribution.

$$f(w_k|K=k) = \frac{\lambda^k}{\Gamma(k)} w_k^{k-1} e^{-\lambda w_k}; \; w_k \geq 0.$$

Note that the distribution of W_k given $K = k$ cannot be exactly gamma, since $P[W'_k \leq nt_0] = 1$ for any k. It follows that $V = 2\lambda W'_k$ can be approximated by a Chi-square variable with $2k$ degrees of freedom. Thus, for example, if $\chi^2_{\alpha/2}$ and $\chi^2_{(1-\alpha/2)}$ are the upper and lower $\alpha/2$ percentages points of the Chi-square distribution with $2k$ degrees of freedom, then

$$\left(\frac{\chi^2_{\alpha/2}}{2W'_k}, \frac{\chi^2_{1-\alpha/2}}{2W'_k}\right)$$

constitutes an approximate $100(1-\alpha)$ percent confidence interval for λ.

3. COMPARISON OF PROCEDURES

A Monte Carlo study was made to compare the three procedures described above. One thousand samples of size n ($n = 30, 40, 50$) from an exponential distribution with parameter λ ($\lambda = 0.1, 0.2, 0.8, 3, 5, 10$) were generated. For each sample, 95-percent confidence intervals for λ were obtained by the three methods (1, 2, 3) for various truncation times, t_0. The results are summarized in Table 1 where we give, for certain combinations of λ, t_0, and method, the average length of the confidence intervals.

the sample variance of these lengths, and the empirical coverage probability (i.e., the proportion of intervals which actually covered λ).

Overall, the procedures appear to rank 2, 3, 1 in decreasing order of general quality of performance. This is clearly the ordering with respect to average interval length, and seems to be the best general ordering with respect to variance in interval length. All three procedures tend to be conservative in terms of coverage probability, with procedure 1 being worst in this respect. Procedure 3 is generally best in terms of more nearly attaining the "target" confidence level $1-\alpha$. Of course such a quality in a procedure is not in itself of value if it is competing with a more conservative procedure which attains comparable (or better) interval length characteristics.

REFERENCES

[1] Halperin, M., Confidence Intervals from Censored Samples, Ann. Math. Stat., **32**, 828–37 (1961).

[2] Rubinstein, D., "Statistical Exposition of Guide Manual for Reliability Measurement Program," NavOrd 29304/Addendum (Nov. 1967).

[3] Wilks, S., Shortest Average Confidence Intervals from Large Samples, Ann. Math. Stat., **9**, 166–175 (1938).

TABLE 1. Some Characteristics of the 95-percent Confidence Intervals Obtained Using Three Procedures

		$n=$	30			40			50		
λ	t_0	PROC.	AVG.	VAR.	C.P.	AVG.	VAR.	C.P.	AVG.	VAR.	C.P.
0.1	3	1	0.150	0.001	0.966	0.136	0.000	0.971	0.120	0.000	0.976
		2	0.146	0.001	0.963	0.127	0.000	0.959	0.112	0.000	0.970
		3	0.157	0.001	0.951	0.132	0.001	0.953	0.116	0.000	0.960
0.1	10	1	0.107	0.000	0.961	0.090	0.000	0.977	0.080	0.000	0.963
		2	0.092	0.000	0.951	0.080	0.000	0.951	0.071	0.000	0.955
		3	0.093	0.000	0.954	0.081	0.000	0.945	0.071	0.000	0.958
0.2	3.2	1	0.251	0.002	0.978	0.200	0.001	0.963	0.176	0.001	0.964
		2	0.213	0.001	0.961	0.183	0.001	0.958	0.163	0.000	0.952
		3	0.217	0.001	0.954	0.186	0.001	0.961	0.165	0.000	0.945
0.8	0.82	1	0.913	0.036	0.966	0.799	0.017	0.959	0.703	0.011	0.961
		2	0.855	0.023	0.942	0.732	0.012	0.948	0.647	0.007	0.945
		3	0.871	0.026	0.936	0.743	0.012	0.954	0.657	0.008	0.936
3	0.11	1	4.51	0.822	0.954	3.84	0.423	0.966	3.43	0.269	0.957
		2	4.23	0.661	0.946	3.60	0.339	0.955	3.22	0.227	0.954
		3	4.56	7.10	0.940	3.73	0.473	0.949	3.32	0.292	0.948
3	0.33	1	5.16	0.426	0.976	2.73	0.227	0.972	2.38	0.122	0.962
		2	2.77	0.196	0.963	2.41	0.113	0.956	2.12	0.066	0.957
		3	2.80	0.222	0.959	2.42	0.118	0.957	2.14	0.071	0.961
5	0.12	1	6.01	1.20	0.974	5.16	0.704	0.962	4.56	0.420	0.957
		2	5.50	0.874	0.951	4.75	0.499	0.955	4.22	0.311	0.955
		3	5.62	1.00	0.951	4.84	0.562	0.950	4.28	0.338	0.955
10	0.06	1	12.1	5.30	0.970	10.2	2.60	0.966	9.07	1.56	0.965
		2	11.1	3.70	0.950	9.39	1.85	0.961	8.40	1.15	0.964
		3	11.1	4.14	0.949	9.55	2.02	0.956	8.54	1.29	0.962
10	0.1	1	10.7	5.47	0.961	9.08	2.62	0.971	7.98	1.63	0.959
		2	9.92	3.46	0.945	7.99	1.30	0.95,	7.08	0.798	0.950
		3	9.43	2.71	0.939	8.05	1.32	0.958	7.12	0.827	0.950

AVG. = Average interval length
VAR. = Sample variance in interval lengths
C.P. = Empirical coverage probability

*U.S. GOVERNMENT PRINTING OFFICE: 1972-010-497/3