AN ABSTRACT OF THE THESIS OF

Fernando L. Taracena for the degree of Doctor of Philosophy in Industrial Engineering presented on April 20, 1983

Title: A Bayesian Analysis for Economic Design of Single Sampling Plans for a Sequence of Lots

Abstract approved: Redacted for Privacy
Edward D. McDowell

A dynamic approach to the Bayesian theory of sampling inspection by attributes for the single sampling case is presented. A model is developed which assumes a sequence of lots of equal known size and lots generated by a process operating in a random manner. The model also assumes constant associated costs for all lots and that the posterior distribution of the process quality of a given lot becomes the prior distribution of the process quality of the next lot. This allows the information from one lot to be used in the decision making for subsequent lots. The model is formulated by using a mixed binomial distribution with beta weights as the prior distribution of the lot quality.

An improved algorithm for the solution of the single lot case, bounds for the optimal sample size, a lower bound for the expected cost of sampling for the single lot case and a lower bound for the

expected cost for the sequence of lots are presented. Optimality conditions for the non-sampling alternatives, the 100% sampling case, and the convergence of the optimal acceptance plan when the number of lots in the sequence tends to infinity are investigated. The model is formulated as a dynamic programming problem with sampling, reject without sampling and accept without sampling as the possible actions; and the lots as the stages. Relationships between the optimal actions at different lots which are independent of the form of the expected cost of sampling function are presented. Exact and approximate solution algorithms are developed and tested.

Experimental results indicate that the use of the bounds for the optimal sample size, the lower bound for the expected cost of sampling and the results on optimality of the non-sampling alternatives lead to the pruning of a large part of the decision tree.

Approximate methods developed can be classified as forward-backward procedures. The forward pass reduces the state space by fixing the sample size for all lots but the last one, according to a specified set of rules. The backward pass uses the dynamic programming formulation for finding the optimal policy for the reduced state space. The effectiveness of the approximate methods were evaluated in terms of the quality of the solutions and the computational effort to obtain the solutions. Results indicate that efficiency of two of the approximate methods is very high while computational requirements are drastically reduced.

A Bayesian Analysis for Economic Design of Single Sampling Plans for a Sequence of Lots

by

Fernando L. Taracena

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed April 20, 1983

Commencement June 1983

APPROVED:

Redacted for Privacy

Professor of Industrial Engineering in charge of major

Redacted for Privacy

Head of department of Industrial Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented April 20, 1983

Typed by Sadie Airth for Fernando L. Taracena

ACKNOWLEDGEMENTS

I am deeply indebted to my major professor Dr. Edward McDowell who perceived the need for this research and for his support , guidance and enthusiasm during the full course of this study.

Special thanks to Drs. Jeff Arthur and David Faulkenberry for their advice and outstanding courses which provided the basic tools for this research.

To Drs. Eugene Fitcher and James Riggs who provided finantial support in the form of an assistantship.

In addition, I would like to thank the University Computer Center that supplied the computing services that made possible the completion of this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

A BAYESIAN ANALYSIS FOR ECONOMIC DESIGN OF SINGLE SAMPLING PLANS FOR A SEQUENCE OF LOTS

CHAPTER 1.

INTRODUCTION

1.1 Background

Since the work of Dodge and Romig (1929), acceptance sampling has gained wide application in industry, and it is one of the major fields of statistical quality control. Since their introduction, acceptance sampling plans have been designed with respect to statistical criteria. However, in recent years the emphasis in acceptance sampling has moved steadily from classical to Bayesian methods.

The Dodge-Romig models are based on the assumption that the process quality level, p, is known with certainty, and therefore the only reason that motivates inspection is the possibility of outliers. In practice, p fluctuates and therefore, it is more realistic to consider p as a random variable with an associated distribution function.

The lot quality level, p^1, has two elements of randomness, the process quality, p, and the random fluctuation of p^1 about p. Bayesian methods assign a probability distribution, called a prior, to the process quality level p, and therefore the models are

specified by the joint distribution of the observed number x of defectives in the sample and the random variable p. The probability distribution of the process quality is called the prior, because it represents our belief or information about the random variable p prior to the experiment.

The importance of the prior distribution was pointed out by Mood (16), who concluded that the number of defectives, x, found in a sample of size n, provides no information about the number of defectives left in the lot, y, when p is a known constant.

Bayesian methods have been applied to single sampling, double sampling, multiple sampling, and sequential sampling. However, as Hald (13) p. 34 pointed out;

> The present theory is formulated as if the problems were static, the only exception being the rules for switching between normal and tightened inspection. In practice the prior distribution may shift, the frequency of outliers may increase and there exists an interaction between the system of sampling inspection used and the prior distribution. What we need is a dynamic theory with a feedback mechanism taking these factors and the information from previous inspection results into account.

Standard military sampling procedures for inspection by attributes use three types of inspection; normal, tightened and reduced. The switching procedures between types of inspection are a function of the inspection results of the preceding lots, i.e. the decision of accepting or rejecting the current lot is affected by the results of previous inspections. The Militar Standard 105D has as it's main purpose the preservation of the acceptance quality level or AQL,

which the producer and consumer have agreed upon. No economic criteria nor Bayesian methods are considered.

1.2 Purpose and Scope of the Research

The theory of sampling inspection was originally developed to control the quality of lots or batches of mass-produced articles in industry. The most important aspect being to determine a course of action regarding disposition of the lot. It is usually assumed that we have to choose between three courses of action: (1) Acceptance without sampling inspection; (2) Rejection without sampling inspection; and (3) Sampling inspection followed by either acceptance or rejection.

It is important to note that 100% inspection is considered a special case of the sampling alternative when the sample size is equal to the lot size. The decisions to accept or reject are called terminal decisions. Traditional sampling plans are designed to meet a predetermined producer's and consumer's risk. The producer is interested in having good quality lots accepted and therefore he desires a high probability for acceptance for good lots. On the other hand, the consumer is interested in rejecting bad quality lots, and therefore he desires a low probability of acceptance for bad lots. Clearly, the definition of good and bad lots depends on the particular circumstances. This probability of acceptance considered as a function of the quality of the lot is called the operating characteristic function (OC Curve) of the sampling plan.

Bayesian models for sampling inspection plans commonly assess the costs and losses involved in operating a given plan and try to minimize total costs. If the size of the sample is increased, the losses from wrong decisions are reduced, but clearly the cost of sampling is increased. Hence there is an economic optimum which can be determined.

The main objective of this research is to expand the present Bayesian theory of sampling inspection by attributes from a static to a dynamic approach, by developing a dynamic model with a feedback mechanism to use the information from previous inspection results in the decision making for the current lot.

To include the information from previous inspection results, a sequence of lots coming from a process which operates in a random manner is considered. The costs and losses involved in operating a policy for the sequence of lots are estimated and the total expected cost of the whole sequence is minimized. Now, if the size of the sample of say, the first lot is increased, the losses from wrong decisions in that lot are reduced and, in addition, the information for subsequent lots is increased and therefore, losses from wrong decisions for those lots are also reduced.

The model formulated should allow us to recognize the learning process incurred by sampling a lot and passing that information to the next. In addition, it should have sufficient flexibility to adequately represent a real situation.

The objective we will use is the minimization of the total expected cost, which is, of course, a very limited objective. It is important to notice that in practice, a sampling plan is often required to serve other purposes as well. For example, by using an acceptance plan, quality of production is usually improved through its encouragement of good quality by a high rate of acceptance and its discouragement of poor quality by a high rate of rejection. However, in most cases all indirect effects of sampling inspection are very difficult to quantify.

Proposed solutions known by the author for the single lot case are computationally inefficient for the proposed sequence of lots. Therefore, an improved algorithm to the best known is presented. A lower bound for the expected cost of sampling and necessary and sufficient conditions for optimality are proposed. Some general results which do not depend on the form of the expected cost of sampling are developed.

A major concern of the research is the capability of the model to solve problems of practical dimensions, that is, to be able to solve industrial type problems in a reasonable amount of computer time. Therefore, an approximate solution to the problem is proposed. Measures of performance used are quality of solution and computational effort.

1.3 General Definitions and Terminology

In sampling inspection by attributes the product is divided into inspection lots. The individually inspected part of an inspection lot will be called an item. The size of an inspection lot is the number of items in the lot.

A defect is any non-conformance to some specified requirements. An item which contains one or more defects will be considered a defective item.

The quality of a lot is defined as the number of defectives divided by the number of items in the lot, i.e., fraction defective and the quality of a lot will be used interchangeably.

It is assumed that the inspection procedure is 100% effective, so that inspected items are always classified correctly. This assumption is not always true in practice.

A sample is a group of items selected individually at random from the inspection lot without replacement.

The term rejection is used collectively for all the possible actions taken on lots which are not accepted. A rejected lot may need to be returned to the supplier or, in what is called "rectifying inspection", to totally inspect the lot and defective items corrected or replaced by good ones. Other possibilities exist, for example, scrapping, reworking, downgrading, etc.

CHAPTER 2

BAYESIAN SAMPLING PLANS

2.1 State of the Art

Bayesian methods have been applied to acceptance sampling by attributes in most of its forms: single sampling, double sampling, sequential sampling and multiple sampling.

Single sampling plans have received the most analysis; for example, see (6), (7), (8), (9), (10), (13), (15), (18), and (20). Recall that an attribute single sample is characterized by three integers; the lot size, N, the sample size, n, and the acceptance number, c. After inspecting the sample, the lot may be either accepted if the number of defectives found in the sample is c or smaller, or rejected otherwise.

A very significant contribution to the topic was presented by Hald (7). Hald's paper is divided into two main parts. One part is essentially probability theory dealing with the sampling distribution, $g_n(x)$, i.e. the distribution of the number of defects, x, found in a sample of size n before the experiment is performed. This distribution is called by some authors the preposterior distribution. Since $g_n(x)$ is obtained by averaging the hypergeometric distribution for given x over all possible values of the process quality level, Hald calls it the mixed or compound hypergeometric distribution.

Properties of this distribution for several prior distributions are investigated. In the other main part of the paper, a model based on prior distributions and costs is formulated and a general solution is given to the problem of determining the optimum sampling plan, i.e. the plan minimizing the cost function for any prior distribution.

Another important contribution to the theory of acceptance sampling by attributes was made by Raiffa and Schlaifer (20). They developed a Bayesian single sampling plan within the framework of decision theory, looking at the problem in terms of the expected value of information. Although they do not suggest any efficient way of solving for the optimal sample size in the binomial case, they proposed a normal approximation to it.

Most of the models for single sampling have the following cost elements:

a) Cost of sampling,

b) Cost per accepted lot,

c) Cost per rejected lot.

The cost of sampling is usually a linear function of the sample size n, with or without a fixed term, i.e. $k_1 + k_2 n$ where k_2 can be interpreted as the per unit cost of sampling and k_1 a fixed cost incurred when sampling.

The cost per accepted lot is only incurred when x is less than or equal to c. This cost is usually formed by adding to the cost of sampling, the cost of accepting defective items and in some cases, the cost of rejecting the x defectives found in the sample.

Most authors assume that the items sampled are not replaced, but for example, Raiffa and Schlaifer (20) assume the sample, including the x defectives found, is replaced into the lot.

The cost per rejected lot is incurred when x is greater than c. This cost is usually formed by adding to the cost of sampling, the cost of rejecting the non-inspected part of the lot, and in some cases the cost of rejecting the x defectives found in the sample. Here again, most authors assume that items sampled are not replaced.

Methods to obtain the optimal single sampling plans are computationally inefficient and therefore various researchers have proposed approximation methods, see (10), (13), (20), and (22). The most efficient algorithm for finding optimal plans was proposed by Moskowitz and Berry (18). In this paper an upper bound for the sample size is proposed by noting that in order for sampling to be a feasible alternative the cost of sampling has to be less than or equal to the expected value of perfect information (EVPI). The algorithm proposed is based on results given by Raiffa and Schlaifer (20), namely that the expected value of sample information (EVSI) is an increasing function of n, and moreover that the slope of the EVSI between any two contiguous switchover points decreases in n.

Switchover points are defined as the values of n where the value of the c which minimizes the expected total cost, c^*, changes.

Using these results and the additional property that the slope of the EVSI in the neighborhood and to the right of each switchover point is also decreasing in n (this property was only checked

experimentally), the following algorithm is outlined: the switch-over points are used to determine the region of the global optimum by searching for three contiguous switchover points that have the following property: The Expected Total Cost (ETC) of the smallest (n,c^*) is greater than the ETC of the next smallest (n,c^*) which is less than the ETC of the largest (n,c^*), i.e.,

$ETC\ (n,c^*)_i > ETC(n,\ c^*)_j < ETC(n,c^*)_k$, where $n_i < n_j < n_k$ and

$$c_i^* < c_j^* < c_k^*$$

The algorithm is divided into two phases: Phase I finds all switchover points and calculates its ETC and Phase II calculates the ETC for all (n,c^*) in the region with the property above and selects the plan (n^*,c^*) which minimizes the ETC.

This procedure has the deficiency of computing the ETC of every switchover point and the ETC of all sampling plans (n,c^*) within the region of the global minimum. A more efficient way of approaching the problem is to take advantage of the conjectural ditonicity[1] of the ETC(n) evaluated at the switchover points and the conjectural ditonicity of the ETC(n) within two contiguous switchover points, i.e., when c^* does not change. Therefore, an efficient search technique for ditonic functions can be used first to find the switchover points with the property above, and second to find the minimum ETC's in between these switchover points.

Hald (10), (13) obtains an asymptotic solution for continuous prior distributions. He first develops an asymptotic expansion of

1) ditonic function is defined as a discrete function g(j) such that g(j) is decreasing for $1 \leq j \leq j^*$ and increasing for $j > j^*$.

the regret function by means of Taylor Series in terms of h = c/n and n; and then determines the optimum values of h and n by setting the two derivatives equal to zero.

The regret function is defined as the expected cost of sampling, minus the expected cost of the reject and accept without sampling alternatives . As the second term is not a function of n nor c, minimizing the regret function has the same effect as minimizing the expected cost of sampling.

Raiffa and Schlaifer (20) propose an approximation method to the optimal sample size: they found a quick way of obtaining the optimal sample size for the case where the sample observations are normally distributed with known precision and the prior distribution of the process mean is normal. Then they used the results for this special case as an approximation to the optimal sample size in other cases.

Standard search techniques have been applied for finding optimal single sampling plans (see (2) and (23)).

Wetherill and Chiu (26) presented a comprehensive review of the literature on acceptance sampling with emphasis on the economic aspect. They concluded that most researchers assumed the lot quality has a mixed binominal distribution with different weight functions. However, because of mathematical convenience and greater flexibility, the beta weight function is usually preferred.

The theory of Bayesian single sampling has been generalized to double sampling (see (12), (13), (19), and (24)). Recall that a double sampling plan specifies two sample sizes, n_1 and n_2, two

acceptance numbers, c_1 and c_2, and two rejection numbers, r_1 and $r_2 = c_2 + 1$. A sample of size n_1 is taken and if c_1 items or less are defective, the lot is immediately accepted. If r_1 or more items are defective, it is rejected. Otherwise, a sample of n_2 items is taken. Then, if the total number of defective items is c_2 or less, the lot is accepted, otherwise the lot is rejected. Stewart, Montgomery and Heikes (24) presented an approximate model for the optimum economic design of double sampling plans using the beta distribution as prior. They expressed the expected total cost function as a sum of integrals and evaluated the function numerically, then a pattern search technique was used to optimize it. Sequential sampling plans treated in the literature have been of the item-by-item type. That is, following each item inspected there is a decision to either accept the lot, reject the lot, or inspect another item. The decision is usually made according to a set of rules derived from Wald's Sequential Probability Ratio Plan.

Numerous papers have been written about sequential sampling plans, for example, see (3), (14), and (27). Lindley and Barnett (14) used the beta-binomial model and derived relationships for finding the sequential sampling accept-reject boundaries in terms of the parameters of the beta distribution. Wortham and Wilson (27) presented a computer oriented approach which provides flexibility in the choice of the model and the prior distribution. It is important to note that in all cases optimal solutions were found through dynamic programming type procedures.

Recently, interest has been directed to the economic analysis of multiple sampling procedures (also known as multistage procedures). See (1), (11), (17), and (21).

Multiple sampling is an extension of the double sampling scheme just described, and they can have fixed or variable sample sizes. In the variable sample sizes case, at each stage of the procedure, the size of the next sample is a function of the outcome of all the samples up to that point, and similarly when the size of the next sample is zero or when we are at the final stage, the decision of whether the lot is accepted or rejected is prescribed in terms of the sampling outcome thus far obtained. If all sample sizes are fixed in advance, then it is only necessary to determine at every stage whether the procedure is going to be continued; and if not, what terminal decision (accept or reject) is to be made.

Schuler (21) obtains a general result for both cases which, however, are of more theoretical than practical interest. Hald and Keiding (11) worked with a simpler model and developed asymptotic solutions for both problems. Their model assumes a differentiable prior distribution, a linear loss function, an asymptotically normal sampling distribution and sampling costs proportional to the sample size. The resuls generalize previous known results for single sampling and double sampling.

Moreno (17) presents a multiple sampling approach with fixed sample sizes, which he reports has been tested successfully since 1971 in an industrial environment. He uses the expected outgoing

quality and the expected quality assurance cost as measures of performance. The model is structured as a Markov Decision Problem with transition probabilities between sampling stages given by Bayesian estimates. The optimal solutions are obtained through dynamic programming with constraints to represent contractual restrictions, and with directed steps to evaluate given sampling plans.

A different problem which, however, gives us interesting results for our project was presented by Dietrich and Sanders (4). They developed a systematic method of determining the sampling policy throughout a multistage production process by using economic criteria. That is, to find at every stage of the process whether you should sample, inspect or reject the batch, and in the case that sampling is optimal to find the optimal single sampling plan. The model uses the mixed binomial with beta weights as prior distribution and the sampling outcome is used to update the parameters of the prior distributions.

2.2 The Model for a Single Lot

2.2.1 Notation. Let the lot size be denoted by N where N=1,2,... . The number of defectives in the lot by X; X=0,1,..., N. and the fraction defective by $p^1 = X/N$.

The sample size will be denoted by n: n=1,2,...,N. Occasionally we will use n=0, meaning that sample was not taken. The number of defectives in the sample will be denoted by x: x=0,1,...,n.

As mentioned previously, a single sampling plan is defined by means of three numbers (N,n,c), where c denotes the acceptance number; c=-1,0,1,...,n. The inclusion of c=-1 is for notational convenience. Since $0 \leq x \leq n$, the cases c=-1 and c=n leads us to two singular cases. c=-1 means reject the lot regardless of the outcome of the sample; and c=n means accept the lot regardless of the outcome of the sample.

2.2.2. Cost Structure. The model will contain the following cost elements: cost of sampling, cost per accepted lot and the cost per rejected lot.

Costs of sampling include sampling and testing costs. Defining k_I as the fixed cost of sampling inspection per lot and k_v as the variable cost of sampling and testing one item for presence of all attributes; then the cost of sampling a lot is given by k_I+k_vn. Note that k_I represents all direct and indirect costs that result from sampling and testing and are independent of the sample size; and k_v represents all direct costs associated with the inspection of an item. We define k_r as the unit cost of the action taken on rejected items, where, as we mentioned before, the action taken on rejected items may mean they are returned to supplier, rectifying inspection or reworking, etc. Then, for example, the cost of rejecting a lot without inspection is given by k_rN.

The quantity most difficult to determine is normally the cost of accepting a defective item, which will be denoted by k_a. If the items considered are to be further processed, then the cost of

accepting a defective item may, for example, consist of the cost of handling and identifying the defective item, damage done to other items, cost of rework or cost of replacing it and costs of renewed testing and inspection. If the items are finished goods, the cost of passing a defective item may involve service and replacement costs plus loss of goodwill, which is difficult to measure.

2.2.3 Alternatives to Sampling. We consider two alternatives to maintaining a sampling plan: accepting and rejecting without inspection. The cost of accepting a lot without inspection is CAWI = $k_a X$ and the cost of rejecting a lot without inspection is CRWI=$k_r N$.

If CAWI is greater or smaller than CRWI for any possible fraction defective (p^1), then the problem becomes trivial, as we know the optimal action to take, regardless of the true fraction defective of the lot. Therefore, we shall assume that CAWI and CRWI intersect at $p^1 = p_c$ (critical fraction defective or breakeven quality) where $0 < p_c < 1$. As by solving the above equations $p_c = k_r/k_a$, we shall in consequence assume that $\infty > k_a > k_r$, which in most cases is true. p_c is usually called the critical quality level or break-even quality level since when the true p^1 is less than p_c, it is cheaper to accept than to reject, and when the true p^1 is greater than p_c, the opposite is true. Clearly, at $p^1 = p_c$ both alternatives have the same cost.

If we assume $k_v \geq k_r$ to avoid the possibility of 100% sampling being the optimal policy and if we assume the true p^1 is known, then we are able to choose the correct alternative. The minimum cost per item for lots of known quality, k^*, is given by

$$k^* = \begin{cases} k_a p^1 & ; \ p^1 \leqslant p_c \\ k_r & ; \ \text{otherwise} \end{cases} \qquad (2.1)$$

k^* gives the cost per item when we make the right decision without sampling. However, we do not know p^1, and in order to get information about it, we may use sampling inspection.

2.2.4 Expected Cost of Sampling. In sampling inspection; the costs associated with lots of quality $p^1 = X/N$ will be composed of two terms.

a) The cost per accepted lot.

The cost per accepted lot is formed by the cost of sampling plus the cost of accepting X-x defectives which are left in the lot, plus the cost of rejecting the x defectives found in the sample. Defining $y = X-x$ we have:

$$k_I + k_v n + k_a y + k_r x \qquad ; \ 0 \leqslant x \leqslant c \qquad (2.2)$$

b) The cost per rejected lot.

This cost is formed by adding the cost of rejecting the non-sampled items and the x items found defectives, to the cost of sampling, i.e.,

$$k_I + k_v n + k_r(N-n) + k_r x \qquad ; \ c+1 \leqslant x \leqslant n \qquad (2.3)$$

Using (2.2) and (2.3) we get the expected cost of sampling as a function of n, c and X, K (n,c,X), by averaging over all possible x given a specified quality of the lot.

$$K(n,c,X) = \sum_{x=0}^{c}(k_I + k_V n + k_a y + k_r x)\,P(x/X) \;+\; \sum_{x=c+1}^{n}(k_I + k_V n + k_r (N-n+x))\,P(x/X)$$

$$K(n,c,X) = k_I + k_V n + k_r E(x/X) + k_a \sum_{x=0}^{c} yP(x/X) + k_r (N-n) \sum_{x=c+1}^{n} P(x/X) \quad (2.4)$$

where $P(x/X)$ is the probability that a sample of size n contains x defectives given that the lot contains X defectives.

Let $f_N(X);\ X = 0,1,\ldots,N$ be the prior distribution of the lot quality, i.e., $f_N(X)$ denotes the probability that a lot of N items contains X defective items.

To get the expected cost of sampling as a function of n and c, we average K(n,c,X) over all possible values of X according to the prior distribution, i.e.,

$$K(n,c) = \sum_{X=0}^{N} K(n,c,X)\, f_N(X)$$

$$K(n,c) = k_I + k_V n + k_r E(x) + k_a \sum_{X=0}^{N}\sum_{x=0}^{c} yP(x,X) + k_r (N-n) \sum_{X=0}^{N}\sum_{x=c+1}^{n} P(x,X) \quad (2.5)$$

where $P(x,X) = P(x/X)\ f_N(X)$.

But $\sum_{X=0}^{N}\sum_{x=0}^{c} yP(x,X) = \sum_{X=0}^{N}\sum_{x=0}^{c} yf(X/x)\, g_n(x) = \sum_{x=0}^{c} g_n(x)\, E(y/x)$

where $g_n(x)$; $x = 0,1,\ldots,n$ is the sampling distribution; the distribution of the number of defects, x, found in a sample of size n before the experiment is performed. $E(y/x)$ is the expected number of defectives left in the lot given that we found x defectives in the sample.

In addition, $$\sum_{X=0}^{N}\sum_{x=c+1}^{n} P(x,X) = \sum_{x=c+1}^{n} g_n(x)$$

Inserting these results in (2.5) we get

$$K(n,c) = k_I + k_V n + k_r E(x) + k_a \sum_{x=0}^{c} g_n(x) E(y/x) + k_r (N-n) \sum_{x=c+1}^{n} g_n(x) \qquad (2.6)$$

Note that $y/(N-n)$ gives the fraction defective in the non-inspected part of the lot. If we define $p^* = y/(N-n)$ then $E(p^*/x)$ is the expected fraction defective in the non-inspected part of the lot, given that x defectives were found in the sample. Using this definition in equation (2.6) we get

$$K(n,c) = k_I + k_V n + k_r E(x) + k_a (N-n) \sum_{x=0}^{c} g_n(x) E(p^*/x) + k_r (N-n)\left(1 - \sum_{x=0}^{c} g_n(x)\right)$$

By defining $$F(n,c) = k_a \sum_{x=0}^{c} (E(p^*/x) - p_c) g_n(x) \qquad (2.7)$$

and rearranging terms we find

$$K(n,c) = k_I + k_V n + k_r (E(x) + N - n) + (N-n) F(n,c) \quad ;n=1,..N \quad c=-1,0,1,..,n \qquad (2.8)$$

The expected cost of sampling, K(n,c) is composed of two elements:

(1) The cost of sampling, CS(n), where

$$CS(n) = k_I + k_V n \tag{2.9}$$

(2) And the expected cost of sampling given a sample of size n=1,..,N has been taken

$$k_r(E(x)+N-n) + (N-n)F(n,c) = K(n,c) - CS(n) \tag{2.10}$$

When c=n, we accept the lot regardless of the outcome of the sample. From equation (2.2)

$$K(n,c=n) = \sum_{X=0}^{N}\sum_{x=0}^{n}(k_I + k_V n + k_a(X-x) + k_r x)P(X,x)$$

$$K(n,c=n) = k_I + k_V n + k_a(E(X)-E(x)) + k_r E(x) \tag{2.11}$$

We arrive at the same result from equation (2.8) by noting that

$$F(n,c=n) = k_a(E(X)-E(x))/(N-n) - k_r = k_a E(p) - k_r$$

When c=-1, we reject the lot regardless of the outcome of the sample. From equation (2.3)

$$K(n,c=-1) = \sum_{X=0}^{N}\sum_{x=0}^{n}(k_I + k_V n + k_r(N-n) + k_r x)P(X,x)$$

$$K(n,c=-1) = k_I + k_V n + k_r(N-n) + k_r E(x) \tag{2.12}$$

which is the same as equation (2.8) with F(n, c=-1)=0.

Mood (16), concluded that when the correlation between the number of defective items in the sample, x, and the number of defective items in the remainder of the lot, y, is positive, then the acceptance criterion for the inspection plan is:

Accept when $x \leq c$; reject otherwise.

However, when the correlation between x and y is zero, sampling provides no information about the non-inspected part of the lot. When this correlation is negative, a large number of defectives in the sample indicates the opposite about the remainder of the lot. Therefore, the acceptance criterion for the inspection plan has to be reversed, i.e.,

Accept when $x > c$; reject otherwise.

If the correlation between x and y is positive, then the $E(y/x)$ increases with x, and therefore $E(p^*/x)$ increases with x.

In the following we will assume $E(p^*/x)$ is non decreasing in x. From equation (2.8) and by noting that $N-n \geq 0$, the problem of finding the value of c which minimizes $K(n,c)$, considering n fixed, simplifies to:

$$\min_{c} F(n,c) = \min_{c}\left\{\sum_{x=0}^{c}(k_a E(p^*/x) - k_r)g_n(x)\right\} \qquad (2.13)$$

As $k_a > 0$, $k_a E(p^*/x)$ is increasing in x. In addition, $g_n(x) \geq 0$ for all x as it is a probability function. Therefore, to solve (2.13) we need to select all negative terms of the summation. Therefore, the optimal value of c when n is fixed, c_n, is given by

$$c_n = \begin{cases} -1 & ; \text{ if } E(p^*/0) > p_c \\ \text{Max}(x) \text{ such that } x \leq n \text{ and } E(p^*/x) \leq p_c & ; \text{ otherwise} \end{cases} \tag{2.14}$$

An alternative expression for (2.14) are the following set of inequalities:

$$\begin{aligned} & E(p^*/c_n) \leq p_c < E(p^*/c_n+1) && ; \ 0 \leq c_n \leq n-1 \\ & \text{If } E(p^*/0) > p_c && \text{then } c_n=-1 \\ & \text{If } E(p^*/n) < p_c && \text{then } c_n=n \end{aligned} \tag{2.15}$$

Therefore, to find the optimal acceptance number for n fixed, we only need to solve $E(p^*/c_n) = p_c$ for c_n; if we find $0 \leq c_n \leq n$, we use the integer part of c_n; if we find $c_n<0$ we set $c_n=-1$; and if we find $c_n > n$, then we set $c_n=n$.

CHAPTER 3

DYNAMIC MODEL

3.1 Introduction

Consider a sequence of l lots for inspection coming from a process operating in a random manner. We will assume that all lots in the sequence have equal known lot size N, and that the associated costs are constant for all lots. A lot in the sequence has, from Equation (2.8) an expected cost of sampling given by:

$$K(n_i, c_i) = k_I + k_v n_i + k_r (E(x_i) + N-n_i) + (N-n_i) F(n_i, c_i); \quad i = 1,\ldots,l \qquad (3.1)$$

$$n_i = 1,\ldots,N$$

$$c_i = -1,0,1,\ldots,n_i$$

The expected cost of accepting the ith lot without inspection is found by averaging $k_a X_i$ over all possible values of X_i, according to the prior distribution.

$$ECAWI = \sum_{X_i=0}^{N} k_a X_i f_N(X_i) = k_a E(X_i); \quad i=1,\ldots,l \qquad (3.2)$$

The cost of rejecting the ith lot without inspection is a constant given by

$$CRWI = k_r N \qquad (3.3)$$

When considering a single lot we desire to find the minimum cost alternative, i.e. Min $\{K(n,c)$, ECAWI, CRWI$\}$. However, if we use the information of previous sampling results in order to modify the parameters of the current $f_N(X_i)$ and we aim to the expected total cost minimization of the l lots, the current policy may affect the expected cost at later stages of the process.

3.2 Distributions

The probability that a sample of size n contains x defectives given that the lot contains X defectives is given by the probability density function P(x/X). When sampling from a lot of finite size, P(x/X) is hypergeometric, i.e.,

$$P(x/X) = \frac{\binom{n}{x}\binom{N-n}{X-x}}{\binom{N}{X}} = \frac{\binom{X}{x}\binom{N-X}{n-x}}{\binom{N}{n}} ; \qquad \text{Max}(0,n-N+X) \leq x \leq \text{Min}(n,X) \tag{3.4}$$

To select the prior distribution of the lot quality, $f_N(X_i)$, it is necessary to consider some desirable properties.

(i) The prior distribution should allow us to have the $E(p_i^*/x_i)$ non-decreasing in x_i.

(ii) As we are considering a sequence of lots coming from the same process. It is desirable to use the information of past lots in the prior distribution of lot i, i.e. it is desirable to include

(n_1,x_1), (n_2,x_2),..., (n_{i-1},x_{i-1}) in $f_N(X_i)$.

(iii) The prior distribution should belong to a family F with

sufficient flexibility to adecuately represent the process characteristics. That is, F should be rich so that there will exist a member of F capable of expressing the decision maker's prior information and beliefs.

(iv) That the prior distribution preserve its form under sampling and under Bayes' rule. And therefore, the sampling and the posterior distributions be members of the same family F. That is, a closed family (20), page 44.

To find a prior distribution which takes into account the two elements of randomness of the lot quality level, X/N, we start by noting that

$$f_N(X) = \int_0^1 P(X,p)\, dp = \int_0^1 h_N(X/p)\, W(p)dp \qquad (3.5)$$

Where $h_N(X/p)$ represents the fluctuation of X/N about the process quality p, and W(p) represents the pure randomness of the process quality p. W(p) is usually called the prior distribution of the process.

As we are assuming that the process is operating in a random manner, $h_N(X/p)$ follows a binomial distribution and therefore $\int_0^1 h_N(X/p)\, W(p)dp$ is called the mixed binomial distribution with weight W(p).

To choose W(p), we need a class of distributions which concentrate on the interval (0,1) with sufficient flexibility to adequately represent the process characteristics (property (iii)),

one such class is the two parameter beta famly, (r,t). In addition, as the beta is the natural conjugate of the binomial it will satisfy property (iv).

Under this assumption, Equation (3.5) becomes:

$$f_N(X)=\binom{N}{X}\int_o^1 p^X q^{N-X} W(p)dp \quad ; X=0,1,..,N \quad ; 0<p<1 \quad ; q=1-p \tag{3.6}$$

where: $W(p)= p^{r-1}q^{t-r-1}/B(r,t)$; $0<p<1$; $t>r>0$

$B(r,t)= \Gamma(r)\Gamma(t-r)/\Gamma(t)$; $\Gamma(z)=\int_o^\infty e^{-b}\, b^{z-1}db$, $b>0$

with $E(X) = Nr/t$ and $V(X) = Nr(t-r)(1+N/t)/(t(t+1))$

The mixed binomial distribution with beta weights, Equation (3.6) is called the beta-binomial distribution. Hald (13) pp. 131-133 discusses how to find out whether the beta-binomial distribution is applicable to a specific problem by the analysis of past inspection data.

In order to satisfy property (ii) we will assume that the posterior distribution of the process quality, p, of lot i-1, becomes the prior distribution of p for lot i , where i=2,...,l. That is,

$$W_1(p/x_1)= W_2(p);\ \cdots\cdots\cdots\ W_{i-1}(p/x_{i-1})= W_i(p);\cdots$$
$$\cdots\cdots\ ;\ W_{l-1}(p/x_{l-1})= W_l(p) \quad ; \quad 0<p<1 \tag{3.7}$$

THEOREM 3.1

$$W_{i+1}(p)= W_i(p/x_i)= \frac{p^{r_i-1}\, q^{t_i-r_i-1}}{B(r_i\ ,\ t_i)} \quad ; 0<p<1 \quad ; t_i>r_i>0 \quad ; \text{for } i=1,..,l-1 \tag{3.8}$$

Where: $r_i = r_o + \sum_{j=1}^{i} x_j$ and $t_i = t_o + \sum_{j=1}^{i} n_j$, and r_o and t_o are the initial parameters of the beta distribution. That is,

$W_i(p/x_i) \sim \beta(r_i, t_i)$; beta with parameters r_i and t_i, and the sampling distribution, $g_{n_i}(x_i)$, is mixed binomial with weight function $W_i(p)$ for $i=1,\ldots,l$

$$g_{n_i}(x_i) = \binom{n_i}{x_i} \int_o^1 p^{x_i} q^{n_i - x_i} W_i(p)\, dp \quad ; \; x_i = 0, 1, \ldots, n_i \quad ; \; 0 < p < 1 \quad ; \; \text{for } i = 1, \ldots, l \tag{3.9}$$

Proof:

By induction for $i=1$

$$W_1(p) = p^{r_o - 1} q^{t_o - r_o - 1} / B(r_o, t_o) \quad ; \; 0 < p < 1 \quad ; \; t_o > r_o > 0$$

$$h(x_1/p) = \binom{n_1}{x_1} p^{x_1} q^{n_1 - x_1} \quad ; \; x_1 = 0, 1, \ldots n_1 \quad ; \; 0 < p < 1$$

$$P(x_1, p) = h(x_1/p) W_1(p) = \binom{n_1}{x_1} p^{r_1 - 1} q^{t_1 - r_1 - 1} / B(r_o, t_o) \; ; t_1 > r_1 > 0 \tag{3.10}$$

$$g_{n_1}(x_1) = \int_o^1 P(x_1, p)\, dp = \binom{n_1}{x_1} B(r_1, t_1) / B(r_o, t_o)$$

By using Bayes theorem

$$W_1(p/x_1) = \frac{P(x_1, p)}{g_{n_1}(x_1)} = \frac{p^{r_1 - 1} q^{t_1 - r_1 - 1}}{B(r_1, t_1)} \quad ; \text{ that is, } W_1(p/x_1) \sim \beta(r_1, t_1)$$

From (3.10)

$$g_{n_1}(x_1) = \binom{n_1}{x_1} \int_o^1 \frac{p^{r_1 - 1} q^{t_1 - r_1 - 1}}{B(r_o, t_o)}\, dp = \binom{n_1}{x_1} \int_o^1 p^{x_1} q^{n_1 - x_1} W_1(p)\, dp \; ; \; x_1 = 0, 1, \ldots, n_1$$

Assume $i=h$ holds. For $i=h+1$

from (3.7) $W_{h+1}(p)=W_h(p/x_h)= p^{r_h-1}q^{t_h-r_h-1}/B(r_h,t_h)$; $t_h>r_h>0$

$$h(x_{h+1}/p)=\binom{n_{h+1}}{x_{h+1}}p^{x_{h+1}}q^{n_{h+1}-x_{h+1}} \quad ; \quad x_{h+1}=0,1,..,n_{h+1}$$

$$P(x_{h+1},p)=\binom{n_{h+1}}{x_{h+1}}p^{r_{h+1}-1}q^{t_{h+1}-r_{h+1}-1}/B(r_h,t_h) \quad ; \quad t_{h+1}>r_{h+1}>0 \qquad (3.10a)$$

$$g_{n_{h+1}}(x_{h+1})=\int_0^1 P(x_{h+1},p)\,dp=\binom{n_{h+1}}{x_{h+1}}B(r_{h+1},t_{h+1})/B(r_h,t_h)$$

By using Bayes Theorem

$$W_{h+1}(p/x_{h+1})=\frac{p^{r_{h+1}-1}q^{t_{h+1}-r_{h+1}-1}}{B(r_{h+1},t_{h+1})} \quad , \text{ That is, } W_{h+1}(p/x_{h+1})\sim\beta(r_{h+1},t_{h+1})$$

From (3.10a)

$$g_{n_{h+1}}(x_{h+1})=\binom{n_{h+1}}{x_{h+1}}\int_0^1 p^{x_{h+1}}q^{n_{h+1}-x_{h+1}}W_{h+1}(p)\,dp \quad ; \quad x_{h+1}=0,1,..,n_{h+1}$$ ■

Recalling (3.6), the lot quality prior distribution of lot i, $f_N(X_i)$ can be written as

$$f_N(X_i)=\binom{N}{X_i}\int_0^1 p^{X_i}q^{N-X_i}W_i(p)\,dp \quad ; \quad X_i=0,1,..,N \quad ; \quad 0<p<1 \qquad (3.11)$$

where $W_i(p)= W_{i-1}(p/x_{i-1})\sim\beta(r_{i-1},t_{i-1})$

Note that the information of all past lots is included in the parameters of the beta distribution and therefore satisfy property (ii).

Theorem 3.2

The posterior distribution of lot i is beta-binomial.

$$f(y_i/x_i)=\binom{N-n_i}{y}\int_0^1 p^{y_i}\,q^{N-n_i-y_i}\,W_j(p/x_i)\,dp \quad ; \quad 0<p<1 \qquad (3.12)$$
$$; \quad y_i=0,1,\ldots,N-n_i$$

where $W_j(p/x_i)\sim\beta(r_i,t_i)$

Proof:

By (3.4) and (3.11)

$$P(x_i,X_i)=f_N(X_i)\,P(x_i/X_i)=\binom{n_i}{x_i}\binom{N-n_i}{X_i-x_i}\int_0^1 p^{X_i}q^{N-X_i}\,W_j(p)\,dp \quad ; \quad \begin{matrix}x_i=0,1,\ldots,n_i\\ X_i=0,1,\ldots,N\end{matrix}$$

$$P(x_i,y_i)=\binom{n_i}{x_i}\binom{N-n_i}{y_i}\int_0^1 p^{y_i+x_i}\,q^{N-y_i-x_i}\,W_j(p)\,dp \quad ; \quad y_i=0,1,\ldots,N-n_i$$

Using (3.9)

$$f(y_i/x_i)=\binom{N-n_i}{y_i}\int_0^1 p^{y_i}q^{N-n_i-y_i}\,p^{x_i}q^{n_i-x_i}\,W_j(p)\,dp\Big/\int_0^1 p^{x_i}q^{n_i-x_i}\,W_j(p)\,dp$$

But note that

$$\frac{p^{x_i}q^{n_i-x_i}\,W_j(p)}{\int_0^1 p^{x_i}\,q^{n_i-x_i}\,W_j(p)\,dp}=\frac{k(x_i/p)W_j(p)}{\int_0^1 k(x_i/p)W_j(p)\,dp}=\frac{P(x_i,p)}{g_{n_i}(x_i)}=W_j(p/x_i)$$

$$f(y_i/x_i)=\binom{N-n_i}{y_i}\int_0^1 p^{y_i}q^{N-n_i-y_i}\,W_j(p/x_i)\,dp \quad ; \quad \begin{matrix}y_i=0,1,\ldots,N-n_i\\ 0<p<1\end{matrix} \qquad (3.13)$$

■

From Theorems 1 and 2 we can see that by choosing $f_N(X_i)$ mixed binomial, property (iv) is satisfied. That is, the sampling and posterior distributions are members of the same family. From theorem 1 and by noting that $r_i = r_{i-1} + x_i$ and $t_i = t_{i-1} + n_i$ we get

$$g_{n_i}(x_i)=\binom{n_i}{x_i}\,B(r_i,t_i)/B(r_{i-1},t_{i-1}) \quad ; \quad x_i=0,1,\ldots,n_i$$

$$g_{n_i}(x_i)=\binom{n_i}{x_i}\Gamma(r_i)\,\Gamma(t_i-r_i)\,\Gamma(t_{i-1})/\left(\Gamma(t_i)\,\Gamma(r_{i-1})\,\Gamma(t_{i-1}-r_{i-1})\right) \qquad (3.14)$$

From the relationship $\Gamma(p+k)= (p+k-1)!\Gamma(p)/(p-1)!$ we obtain

$\Gamma(t_i)=(t_i-1)!\Gamma(t_{i-1})/(t_{i-1}-1)!$; $\Gamma(r_i)=(r_i-1)!\Gamma(r_{i-1})/(r_{i-1}-1)!$

and

$$\Gamma(t_i-r_i)=(t_i-r_i-1)!\Gamma(t_{i-1}-r_{i-1})/(t_{i-1}-r_{i-1}-1)!$$

Inserting this in (3.14) we obtain

$$g_{n_i}(x_i)=\frac{\binom{n_i}{x_i}\frac{(r_i-1)!}{(r_{i-1}-1)!}\frac{(t_i-r_i-1)!}{(t_{i-1}-r_{i-1}-1)!}}{\frac{(t_i-1)!}{(t_{i-1}-1)!}} = \frac{\binom{r_i-1}{x_i}\binom{t_i-r_i-1}{n_i-x_i}}{\binom{t_i-1}{n_i}} \qquad (3.15)$$

The computation of $g_{ni}(x_i)$ is greatly simplified by the use of recursive relationships. By means of (3.14) and (3.15) we obtain the following relations.

$$g_{n_i}(x_i)=(r_{i-1}+x_i-1)(n_i-x_i+1)g_{n_i}(x_i-1)/(x_i(n_i+t_{i-1}-r_{i-1}-x_i)) \; ; \; x_i=1,2,..,n_i \qquad (3.16)$$

$$g_{n_i}(x_i)=n_i(t_{i-1}-r_{i-1}+n_i-x_i-1)g_{n_i-1}(x_i)/((n_i-x_i)(t_{i-1}+n_i-1)) \; ; \; x_i=0,1,..,n_i \; ; \; n_i \geq x_i \; ; \; n_i=1,2,..,N \qquad (3.17)$$

In addition, from (3.14) we obtain

$$g_{n_i}(0)=\Gamma(t_{i-1}-r_{i-1}+n_i)\,\Gamma(t_{i-1})/(\Gamma(t_i)\,\Gamma(t_{i-1}-r_{i-1}))$$

And by using the fact that $\Gamma(p+1)= p\Gamma(p)$

$$g_{n_i}(0)= \frac{(t_{i-1}-r_{i-1}+n_i-1)(t_{i-1}-r_{i-1}+n_i-2)\cdots\cdots\cdots(t_{i-1}-r_{i-1})}{(t_{i-1}+n_i-1)(t_{i-1}+n_i-2)\cdots\cdots\cdots(t_{i-1})} \qquad (3.18)$$

When computing the mixed binomial we use Equation (3.18) to find an initial value and then Equations (3.16) and (3.17).

Theorem 3.3

$$E(p_i^*/x_i) = r_i/t_i = (r_0+\sum_{j=1}^{i} x_j)/(t_0+\sum_{j=1}^{i} n_j) \quad ; \; t_i > r_i > 0 \tag{3.19}$$

And therefore is non-decreasing in x_i, Property (i), and non-increasing in n_i

Proof:

$E(p_i^*/x_i) = E(y_i/x_i)/(N-n_i)$; by Theorem 2, $f(y_i/x_i)$ is a mixed binomial with weight function $W_i(p/x_i) \sim \beta(r_i,t_i)$; therefore, $E(y_i/x_i) = (N-n_i)r_i/t_i$; and $E(p_i^*/x_i) = r_i/t_i$. ■

From theorem 3 we can find the optimal acceptance number for a fixed sample size. From inequalities (2.15) and equation (3.19).

$E(p_i^*/c_i) = (r_{i-1}+c_i)/(t_{i-1}+n_i) = k_r/k_a$.

Solving for c_i we find $\quad c_i = p_c(t_{i-1} + n_i) - r_{i-1}$ (3.20)

Let define $c'_{n_i} = [c_i]$; where $[A]$ means the greatest integer $\leq A$. The optimal c_i, for fixed n_i, c_{n_i}, is given by

$$\begin{aligned} &\text{If } 0 \leq c'_{n_i} \leq n_i \quad ; \quad c_{n_i} = c'_{n_i} \\ &\text{If } \quad c'_{n_i} < 0 \quad ; \quad c_{n_i} = -1 \\ &\text{If } \quad c'_{n_i} > n_i \quad ; \quad c_{n_i} = n_i \end{aligned} \tag{3.20a}$$

Having defined the optimal value of c_i as a function of n_i, we can state the expected cost of sampling, Equation (3.1) as a function of n_i only. By Theorem 1 $\quad E(x_i) = n_i r_{i-1}/t_{i-1}$.

$$K(n_i) = k_I + k_V n_i + k_r(n_i r_{i-1}/t_{i-1}+N-n_i) + (N-n_i)F(n_i) \; ; \tag{3.21}$$

$$i=1,..,l \; ; \; n_i=1,..,N$$

Where $F(n_i)$ is given by

$$F(n_i)=\sum_{x_i=0}^{c_{n_i}}(k_a(r_{i-1}+x_i)/(t_{i-1}+n_i) - k_r)g_n(x_i) \qquad (3.22)$$

The expected per unit cost of sampling is given by

$$K'(n_i) = K(n_i)/N \qquad (3.23)$$

The expected per unit cost of acceptance without inspection (EUCAWI) is given by

$$EUCAWI = ka\ r_{i-1}/t_{i-1} \qquad (3.24)$$

and the per unit cost of rejecting without inspection (UCRWI) is given by

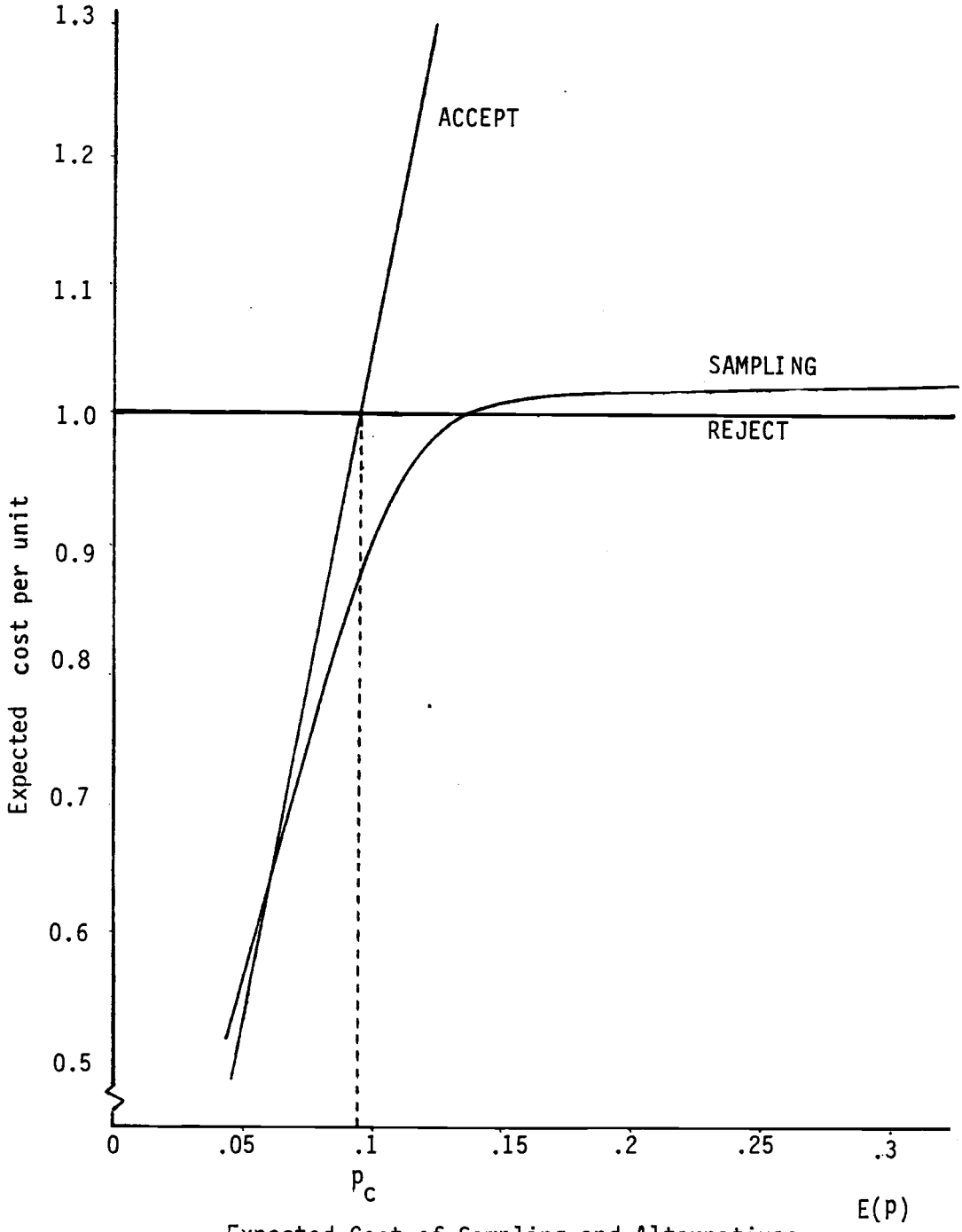
$$UCRWI = kr \qquad (3.25)$$

To illustrate this function, the expected per unit cost of sampling, evaluated at the optimal sample size, $K'(n^*)$ is shown in Figure 1, together with the cost functions of the 2 alternatives to sampling. Equations (3.24) and (3.25) for an example with the following data: N=10, k_I = 0, kv = 0.9, k_r = 1, k_a = 10.5, t = 11. $K'(n^*)$ is evaluated for the following values of r.

Table 1

r	E(p)	n^*	c_n^*	$K'(n^*)$
0.5	0.04545	1	0	0.5113
0.6	0.0545	1	0	0.5913
0.8	0.0727	1	0	0.7469
1.0	0.0909	3	0	0.8627
1.2	0.1091	5	0	0.9395
1.6	0.1454	1	-1	1.0045
1.8	0.1636	1	-1	1.0064
2.0	0.1818	1	-1	1.0082

Optimal sampling plans for Figure 1.



Expected Cost of Sampling and Alternatives.

Figure 1.

Figure 2 shows the behavior of K(n), equation (3.21) for an example with the following data: $N = 50$, $k_I = 0$, $k_v = 0.9$, $k_r = 1$, $k_a = 11$, $r = 2$ and $t = 22$. Note that two local minimum points exist. The global minimum is found at $n^* = 15$ with $c_{n^*} = 1$.

3.3 Analysis of the Single Lot Case

3.3.1 Analysis of K(n)

Let's assume the special case $l=1$. By writing r_o, t_o, n_1 and x_1 as r,t,n and x, respectively (in order to simplify notation); equations (3.21) and (3.22) become

$$K(n)= k_I + k_v n + k_r(n\frac{r}{t} +N-n) + (N-n)F(n) \qquad (3.26)$$

where

$$F(n)= \sum_{x=0}^{c_n}(k_a(r+x)/(t+n) - k_r) g_n(x) \qquad (3.27)$$

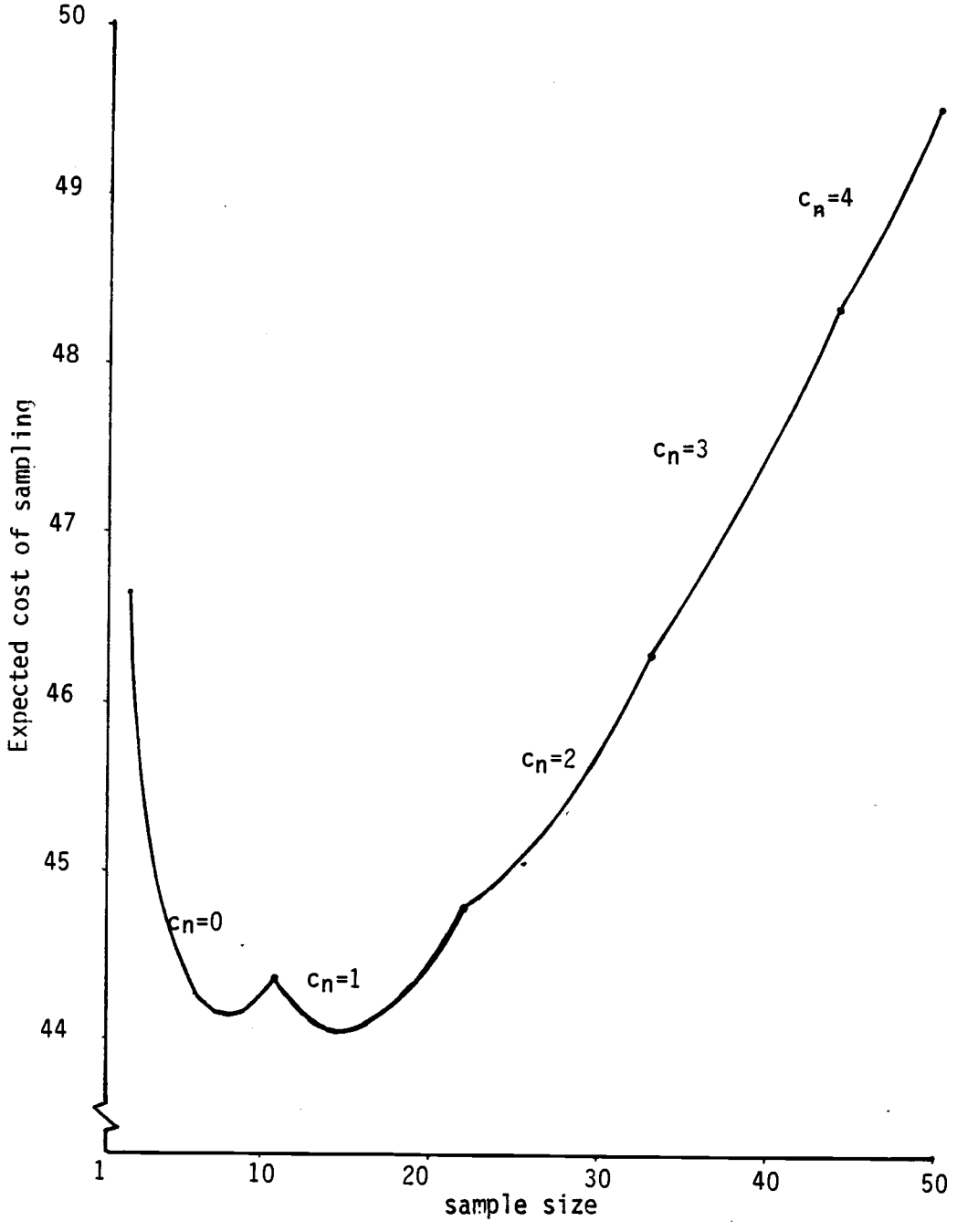
One way of finding the sample size which minimizes the expected cost of sampling, n^*, is to determine the values of n for which $\Delta_n K(n)$ shifts from negative to positive. Where $\Delta_n K(n)$ is defined as K(n+1)-K(n). From equation (3.26).

$$\Delta_n K(n)= k_v + k_r(r/t - 1) + (N-n-1)F(n+1) - (N-n)F(n)$$

$$\Delta_n K(n)= k_v + k_r(r/t - 1) + (N-n)\Delta_n F(n) - F(n+1) \qquad (3.28)$$

From (3.20) $\Delta_n c_n$ can only be 0 or 1. For the case $0 \leq c_n \leq n$.

If $\Delta_n c_n = 0$; then $c_n = c_{n+1}$, and



Expected cost of sampling as a function of the sample size.

Figure 2.

$$\Delta_n F(n)=\sum_{x=0}^{c_n}\Delta_n(k_a E_n(p^*/x) - k_r)g_n(x)$$

$$\Delta_n F(n)= \sum_{x=0}^{c_n}(k_a \Delta_n E_n(p^*/x)g_n(x) - k_r \Delta_n g_n(x)) \qquad (3.29)$$

Hald (7) showed that

$$\sum_{x=0}^{c}\Delta_n E_n(p^*/x)g_n(x)= -g_n(c)E_n(p^*/c)E_{n+1}(p^*/c+1) \qquad (3.30)$$

$$\sum_{x=0}^{c}\Delta_n g_n(x)= -g_n(c)E_n(p^*/c) \qquad (3.31)$$

and $$E_n(p^*/c)= (c+1)g_{n+1}(c+1)/((n+1)g_n(c)) \qquad (3.32)$$

Therefore, Equation (3.29) can be written as

$$\Delta_n F(n)= g_n(c_n)E_n(p^*/c_n)(k_r - k_a E_{n+1}(p^*/c_n+1))$$

$$\Delta_n F(n)= g_n(c_n)(r+c_n)/(t+n)\ (k_r - k_a(r+c_n+1)/(t+n+1))$$

If $\Delta_n\ c_n = 1$; then $c_{n+1} = c_n + 1$

$$\Delta_n\sum_{x=0}^{c_n}E_n(p^*/x)g_n(x)=\sum_{x=0}^{c_n+1}E_{n+1}(p^*/x)g_{n+1}(x) - \sum_{x=0}^{c_n}E_n(p^*/x)g_n(x)$$

$$=\sum_{x=0}^{c_n}\Delta_n E_n(p^*/x)g_n(x) + E_{n+1}(p^*/c_n+1)g_{n+1}(c_n+1)$$

Using (3.30) and (3.32)

$$\Delta_n\sum_{x=0}^{c_n}E_n(p^*/x)g_n(x)= -g_n(c_n)E_n(p^*/c_n)E_{n+1}(p^*/c_n+1)(1 - (n+1)/(c_n+1))$$

And $$\Delta_n\sum_{x=0}^{c_n}g_n(x)=\sum_{x=0}^{c_n}\Delta_n g_n(x) + g_{n+1}(c_n+1)$$

By using (3.31) and (3.32)

$$\Delta_n \sum_{x=0}^{c_n} g_n(x) = -g_n(c_n)E_n(p^*/c_n)(1 - (n+1)/(c_n+1))$$

Therefore

$$\Delta_n F(n) = g_n(c_n)E_n(p^*/c_n)(k_r - k_a E_{n+1}(p^*/c_n+1))(1 - (n+1)/(c_n+1))$$

These results can be summarized as follows:

$$\Delta_n F(n) = \begin{cases} A)\ 0 & ;\ \text{IF } c_n = -1,\ c_{n+1} = -1 \\ B)\ g_{n+1}(0)\left(k_a\, r/(t+n+1) - k_r\right) & ;\ \text{IF } c_n = -1,\ c_{n+1} = 0 \\ C)\ g_n(c_n)\dfrac{r+c_n}{t+n}\left(k_r - k_a \dfrac{r+c_n+1}{t+n+1}\right) & ;\ \text{IF } 0 \le c_n \le n,\ \Delta_n c_n = 0 \\ D)\ g_n(c_n)\dfrac{r+c_n}{t+n}\left(k_r - k_a \dfrac{r+c_n+1}{t+n+1}\right)\left(1 - \dfrac{n+1}{c_n+1}\right) & ;\ \text{IF } 0 \le c_n < n,\ \Delta_n c_n = 1 \end{cases} \qquad (3.33)$$

Defining $\Delta_n^{(2)} F(n) = \Delta_n F(n+1) - \Delta_n F(n)$ for case C) in Equation (3.33) we find

$$\Delta_n^{(2)} F(n) = \frac{g_n(c_n)(r+c_n)}{(n+1-c_n)(t+n)(t+n+1)}\left\{k_r\left[t c_n - (n+1)(r+1)\right] + k_a E_{n+2}(p^*/c_n+1)\left[(n+1)(r+2) - (t+1)c_n\right]\right\}$$

In order to have $\Delta_n^{(2)} F(n) \ge 0$ and therefore convex in the range of n where $0 \le c_n \le n$ and $\Delta_n^{(2)} c_n = 0$, the following condition has to be satisfied in the same range

$$E_{n+2}(p^*/c_n+1)\left(\frac{(t+1)c_n - (n+1)(r+2)}{t c_n - (n+1)(r+1)}\right) \leqslant p_c$$

This condition is usually satisfied but not always, and therefore F(n) is not necessarly convex when c_n is constant.

Theorem 3.4

F(n) is non increasing in n; $1 \le n \le N$.

Proof:

Let $\mathcal{N} = \{n:n=1,\ldots,N\}$. We prove that $\Delta_n F(n) \leq 0$ for $n \in \mathcal{N}$ in all cases of Equation (3.33).

Case a) $\Delta_n F(n) = 0$.

Case b) $\Delta_n F(n) \leq 0$ for $n \in \mathcal{N}$ if and only if $k_r \geq k_a \frac{r}{t+n+1}$ for $n \in \mathcal{N}$.

This follows as $c_{n+1} = 0$ implies by (3.20) that

$$k_r/k_a\ (t+n+1) - r \geq 0 \quad \text{for } n \in \mathcal{N}.$$

Case c) $\Delta_n F(n) \leq 0$ for $n \in \mathcal{N}$ if and only if $k_r \leq k_a \frac{r+c_n+1}{t+n+1}$ for $n \in \mathcal{N}$.

By (3.20) $c_{n+1} + 1 > k_r/k_a(t+n+1)-r$. $\Delta_n c_n = 0$ implies $c_n = c_{n+1}$; therefore condition is satisfied.

Case d) $\Delta_n F(n) \leq 0$ for $n \in \mathcal{N}$ if and only if $k_r \geq k_a \frac{r+c_n+1}{t+n+1}$ for $n \in N$.

By (3.20) $c_{n+1} \leq k_r/k_a\ (t+n+1)-r$. But as $c_{n+1} = c_n + 1$ condition is satisfied.

■

Theorem 3.5

A necessary and sufficient condition for 100% sampling being the cheapest sampling policy is $F(N-1) \geq k_v - k_r\ (1-r/t)$.

Proof:

From Equation (3.26) $K(N) = k_I + k_v N + k_r\ N\ r/t$

$K(N)$ is the minimum expected cost if and only if

$$N(k_v+k_r r/t) \leq k_v n + k_r(nr/t + N-n) + (N-n)F(n) \quad \text{for } n=1,..,N-1$$

or $$F(n) \geq k_v - k_r(1 - r/t) \quad \text{for } n=1,..N-1$$

By Theorem 3.4 $F(n)$ is non-increasing in n, therefore we only need to check for the smallest value of $F(n)$; $F(N-1)$. ■

3.3.2 A Lower Bound for $K(n^*)$

As we discussed before we have three alternatives to choose from: Sample, accept without sampling and reject without sampling. To find the best policy, we have to find the optimal sampling plan. However, if we can construct a lower bound for the optimal expected cost of sampling, LBS, which is easier to compute compared to the effort in finding $K(n^*)$, we may then discard the sampling alternative when $LBS \geq \text{Min}\{ECAWI, CRWI\}$.

Let $D(n)$ be the difference between the expected cost of sampling $K(n)$, $n=1,\ldots,N-1$, and the expected cost of 100% sampling, $K(N)$. From Equation (3.26).

$$D(n) = k_v n + k_r nr/t + k_r(N-n) + (N-n)F(n) - k_v N - k_r Nr/t$$

$$\text{or } D(n) = (N-n)(F(n) - k_v + k_r(1-r/t)) \quad \text{for } n=1,..,N-1$$

If $D(n) < 0$ for some n, $n=1,\ldots,N-1$. 100% sampling is not optimal. By using Theorem 3.4, we conclude that the most negative value that $D(n)$ can take has to be greater than

$$(N-1)\ (F(N-1) - k_v + k_r\ (1 - r/t))$$

Therefore a lower bound for $K(n^*)$ in this case is $K(N)$ plus the quantity above.

If $D(n) \geq 0$ for all n; $n=1,\ldots,N-1$ 100% sampling is optimal. Note that this is true only if $F(n) \geq k_v - k_r\ (1 - r/t)$ for all n; $n=1,\ldots,N-1$, or $F(N-1) \geq k_v - k_r\ (1 - r/t)$ which is the same result given in Theorem 3.5.

Therefore, a lower bound for $K(n^*)$, LBS, is given by

$$LBS=\begin{cases} K(N)+(N-1)(F(N-1)-k_v+k_r(1-r/t) & ; \text{ if } F(N-1)<k_v-k_r(1-r/t) \\ K(N) & ; \text{ otherwise} \end{cases} \quad (3.34)$$

3.3.3 Optimal Policies

In this section we will find the optimal sampling plan for the sampling cases c=n and c= -1 and develop some conditions for which reject and accept without sampling are the optimal policies.

Theorem 3.6

The optimal plan for the sampling case c=n is given by

$n^*=1$, $c^*=1$; if $k_v > E(p)(k_a-k_r)$

100% Sampling ; otherwise

The optimal plan for the sampling case c=-1 is given by

$n^*=1$, $c^*=-1$; if $k_v > k_r(1-E(p))$

100% Sampling ; otherwise

Proof:

From Equation (2.11) $\Delta_n K(n,c=n) = k_v + E(p)\ (k_r-k_a)$. Therefore $\Delta_n K(n,c=n) > 0$ (K(n,c=n) increasing in n) if and only if $k_v > E(p)\ (k_a-k_r)$. K(n,c=n) is non-increasing otherwise. From equation (2.12) $\Delta_n K(n,c= -1) = k_v + k_r\ (E(p)-1)$. Therefore $\Delta_n K(n,c = -1) > 0$ (K(n,c=n) increasing in n) if and only if $k_v > k_r\ (1-E(p))$. K(n,c = -1) is non-increasing otherwise. ■

Theorem 3.7

Reject without sampling is the optimal policy if

$$E(p) > \text{Max}\{p_c(1+N/t) \ , \ 1-k_v/k_r\}$$

Proof:

Lets take the case $c_n = -1$ for all $n \in (1,N)$. This case implies $p_c(t+n) - r < 0$; $1 \leq n \leq N$. This condition is satisfied if and only if

$$p_c(t+N) - r < 0 \quad \text{or} \quad E(p) = r/t > p_c(1+N/t)$$

Which implies $k_a\, r/t > k_r$. That is, the per unit cost of accepting without sampling is greater than the per unit cost of rejecting without sampling.

From Theorem 3.6, the optimal plan for the sampling case $c = -1$ is given by $n^* = 1$, $c^* = -1$ if $E(p) > 1 - k_v/k_r$, therefore for this case the minimum per unit expected cost of sampling is given by (Equation (2.12)).

$$K'(n^*=1, c^*=-1) = (k_I + k_v + k_r(N-1) + k_r E(p))/N$$

$$" \quad > (k_I + k_r(1-E(p)) + k_r(N-1) + k_r E(p))/N$$

$$= (k_I + k_r + k_r(N-1))/N > k_r$$

i.e., the minimum per unit expected cost of sampling is greater than the per unit cost of rejecting without sampling.

■

Theorem 3.8

Accept without sampling is the optimal policy if

$$E(p) < \text{Min}\left\{p_c(1+N/t) - N/t \ , \ k_v/(k_a-k_r)\right\}$$

Proof:

Let's take the case $c_n = n$ for all $n \in (1,N)$. This case implies $p_c(t+n) - r \geq n$; $1 \leq n \leq N$. This condition is satisfied if and only if $p_c(t+N) - r \geq N$ or $E(p) = r/t \leq p_c(1+N/t) - N/t$

or $k_r (1 + N/t) \geq k_a (r/t + N/t)$. This last expression implies $k_r > k_a\, r/t$ as $k_a > k_r$.

From Theorem 3.6. the optimal plan for the sampling case c = n is given by $n^* = 1$, $c^* = 1$ If $E(p) < \frac{k_v}{k_a - k_r}$, therefore for this case the minimum per unit expected cost of sampling is given by (Equation (2.11)).

$$\begin{aligned} K'(n^*=1, c^*=1) &= (k_I + k_v + k_a E(p)(N-1) + k_r E(p))/N \\ &> (k_I + E(p)(k_a - k_r) + k_a E(p)(N-1) + k_r E(p))/N \\ &= (k_I + k_a E(p) N)/N \quad k_a E(p) \end{aligned}$$

■

3.3.4 The Value of Information

The increase in utility which results or would result from taking a sample and learning about the number of defectives in the sample will be called the value of information.

The expected value of sample information (EVSI), is defined as the expected cost of the best decision before sampling minus the expected cost of the best decision given the results of size n. From Equation (3.21).

$$EVSI(n) = \begin{cases} k_a N r/t - (k_r(nr/t + N-n) + (N-n)F(n)) & ; \text{ if } r/t \leq p_c \\ k_r N - (k_r(nr/t + N-n) + (N-n)F(n)) & ; \text{ if } r/t > p_c \end{cases} \quad (3.35)$$

From (3.35) we get the first difference with respect to n

$$\Delta_n EVSI(n) = k_r(1-r/t) - (N-n-1)\Delta_n F(n) + F(n) \; ; \; n=1,..,N-1 \qquad (3.36)$$

By Theorem (3.4) F(n) is non-increasing in n; therefore, $\Delta_n F(n) \leq 0$ for all n, and the first and second terms of Equation (3.36) are greater than zero. However, due to the definition of c_n; $F(n) \leq 0$.

The most negative value that F(n) can take is F(N-1). Therefore, to prove that Δ_n EVSI (n) ≥ 0 for n=1,...,N-1. It is sufficient to prove that

$$\Delta_{N-1} EVSI(N-1) = k_r(1-r/t) + F(N-1) \geqslant 0$$

or $$F(N-1) \geqslant k_r(r/t - 1)$$

This last condition was always satisfied over a wide range of experimental values; corresponding to the fact that we would never refuse to accept free information.

The expected value of perfect information (EVPI) is defined as the expected cost of the best decision before sampling minus the expected cost of the best decision given perfect information about the quality of the lot.

As we assume inspection is 100% effective; perfect information about the quality of the lot is given by 100% inspection of the lot. Therefore, EVPI = EVSI(N). By Equation (3.35)

$$EVPI = \begin{cases} Nr/t(k_a-k_r) & ; \text{ if } r/t \leqslant p_c \\ k_r N(1-r/t) & ; \text{ if } r/t > p_c \end{cases} \qquad (3.37)$$

As the EVSI(n) is non-increasing in n the EVPI $\geq$ EVSI(n) for n=1,...,N-1 and can be interpreted as the maximum potential reduction in total cost of the sampling alternative.

The Expected Net Value of Sample Information (ENVSI) is given by the EVSI minus the cost of sampling.

$$ENVSI(n) = EVSI(n) - CS(n) = Min\ (k_a Nr/t\ ,\ k_r N) - K(n) \qquad (3.38)$$

From the last equality it is clear that if ENVSI(n) > 0, then sampling with a plan (n,c_n) has a smaller expected cost than the non-sampling alternatives, and if ENVSI(n) < 0 for all n, the converse is true.

From Equation (3.38) it may be seen that in order for sampling to be a feasible alternative we require EVPI - CS(n) $\geq$ 0. Therefore, an upper limit on the sample size that should be considered when searching for the optimal acceptance plan is given by

$$k_I + k_v n^* \leqslant EVPI \quad \text{or} \quad n^* \leqslant (EVPI - k_I)/k_v \qquad (3.39)$$

Furthermore, lets assume we found $ENVSI(n_1) > 0$ and therefore, taking a sample of n_1 items has a smaller expected cost than the non-sampling alternatives. We may wonder whether increasing the sample size to n_2 $(n_2 > n_1)$ would give us a better ENVSI or not. In order to increase the ENVSI, or at least maintain it at the same level we have to find that

$$EVSI(n_2) - EVSI(n_1) \geqslant CS(n_2) - CS(n_1) = k_v(n_2-n_1)$$

or

$$n_2 \leqslant n_1 + \frac{EVSI(n_2)-EVSI(n_1)}{k_v} \leqslant n_1 + \frac{EVPI-EVSI(n_1)}{k_v}$$

Therefore a second upper bound for the optimal sample size is given by

$$n^* \leqslant n + \frac{EVPI - EVSI(n)}{k_v} \qquad (3.40)$$

3.4 The Model

3.4.1 Approach

According to our objective, the minimization of the expected total cost of the 1 lots; the optimal solution to our problem is the policy $(n_1^*, c_{n_1^*})$, $(n_2^*, c_{n_2^*}), \ldots, (n_l^*, c_{n_l^*})$, where $0 \leq n_i^* \leq N$ and $-1 \leq c_{n_i^*} \leq n_i^*$ for $i=1,\ldots,l$. Note that we have included the case $n_i^* = 0$ to include the non-sampling alternatives. The policy ($n_i^* = 0$, $c_{n_i^*} = -1$) means rejection without inspection is optimal for lot i and the policy ($n_i^* = 0$, $c_{n_i^*} = 0$) means accepting without inspection is optimal for lot i.

It is important to note here that the policy depends on the actual sampling results. For example, if ($n_1^* = 50$, $c_{n_1^*} = 10$) then $(n_2^*, c_{n_2^*}), \ldots, (n_l^*, c_{n_l^*})$ depend on the outcome of the first sample, x_1.

The problem can be divided into stages, the lots. At each stage there is a policy decision that we have to make: sample, accept without sampling or reject without sampling.

Each lot (stage) has a possible number of states associated with it. The states at stage i are completely defined by the parameters of the beta distribution (r_{i-1}, t_{i-1}). According to the policy decision that we make at any stage, we transform the current state into a state associated with the next stage.

Our process possesses the Markovian property or principle of optimality, that is, given the current state (r_{i-1}, t_{i-1}), the optimal policy for the remaining lots is independent of the policy adopted in previous lots.

Given the properties above, our problem can be formulated and solved as a dynamic programming problem.

In order to understand how the number of states increases with the number of stages and the lot size, let's assume a sequence of three lots of N=2; let's assume further that the first lot is at state (1,10). Then the possible states of lot 2 are: (1,10) if $n_1 = 0$, (1,11) or (2,11) if $n_1 = 1$, and (1,12), (2,12) or (3,12) if $n_1 = 2$. And the possible states of lot 3 are the same as lot 2, plus (1,13), (2,13), (3,13), (4,13), (1,14), (2,14), (3,14), (4,14) and (5,14). That is, we have six possible states at stage 2 and 15 possible states at stage 3.

In general, if we define:

NS_i = Number of possible states in stage i; i=1,2,...,l

$L_i = (i-1)N$; i=1,2,...,l

Then

$$NS_i = (L_i+1)(L_i+2)/2 \quad ; i=1,2,..,l \tag{3.41}$$

Therefore, if we have a sequence of five lots of 1000 items each; then $L_5 = 4000$ and $NS_5 = 8,006,001$.

3.4.2 Formulation

Let the lots i=1,...,l be the stages of our procedure. The tuple (r_{i-1}, t_{i-1}) denotes the possible states of lot i. The actions at any stage i are: sample (S_i), reject without sampling (R_i) and accept without sampling (A_i).

Define $V_i(r,t)$ as the minimum expected cost of lot i and remaining lots i+1,...,l, given that we are currently at state (r,t). Therefore $V_i(r_{i-1},t_{i-1})$ can be described as the minimum expected cost of lot i and remaining lots i+1,...,l, given that we have sampled $\sum_{j=1}^{i-1} n_j$ items and found $\sum_{j=1}^{i-1} x_j$ defectives in lots 1,...,i-1.

The following recursive relationship describes the solution of our problem

$$V_i(r_{i-1},t_{i-1}) = \text{Min}\begin{bmatrix} S_i:\ K(n_i^+) + \sum_{x_i=0}^{n_i^+} V_{i+1}(r_{i-1}+x_i, t_{i-1}+n_i^+) g_{n_i^+}(x_i) \\ R_i:\ k_r N + V_{i+1}(r_{i-1}\ ,\ t_{i-1}) \\ A_i:\ k_a N r_{i-1}/t_{i-1} + V_{i+1}(r_{i-1}\ ,\ t_{i-1}) \end{bmatrix} \quad (3.42)$$

where $n_i^+ \in (1,N)$ $\quad i=1,2,..,l$ and

with boundary conditions $V_{l+1}(\cdot,\cdot) = 0$.

Where $K(n_i^+)$ is given by Equation (3.21), and n_i^+ is the sample size which minimizes $K(n_i) + \sum_{x_i=0}^{n_i} V_{i+1}(r_{i-1} + x_i,\ t_{i-1} + n_i)\ g_{n_i}(x_i)$; $1 \le n_i \le N$. Here it is important to note the difference between n_i^* and n_i^+. n_i^* is the sample size that minimizes $K(n_i)$; $1 \le n_i \le N$; and n_i^+ is as defined above.

Equation (3.42) minimizes the total expected cost of the whole sequence. As all lots have the same size, it is equivalent to minimize the sum of the per unit costs. In addition, as $r_i = r_{i-1} + x_i$; $g_{n_i}(x_i) = g_{n_i}(r_i)$. Therefore, the following relation is equivalent to (3.42)

$$V_i(r_{i-1},t_{i-1}) = \text{Min}\begin{bmatrix} S_i: K(n_i^+) + \sum_{r_i=r_{i-1}}^{r_{i-1}+n_i^+} V_{i+1}(r_i\ ,\ t_{i-1}+n_i^+)g_{n_i^+}(r_i) \\ R_i: k_r + V_{i+1}(r_{i-1}\ ,\ t_{i-1}) \\ A_i: k_a r_{i-1}/t_{i-1} + V_{i+1}(r_{i-1}\ ,\ t_{i-1}) \end{bmatrix} \qquad (3.43)$$

$$\text{where } n_i^+ \in (1,N) \quad \text{and} \quad i=1,2,..,l \quad \text{and}$$

with boundary conditions $V_{l+1}(.,.) = 0$.

3.4.3 Lower Bounds

By using the lower bound for $K(n^*)$ developed in Section 3.3.2 and by using the additional property that $\sum_{r_i=r_{i-1}}^{r_{i-1}+n_i} V_{i+1}(r_i, t_{i-1}+n_i)g_{n_i}(r_i)$ is non-increasing in n_i for $i=1,\ldots,l-1$. This property was only checked experimentally over a wide range of values of the parameters. The following lower bound is proposed for S_i, LBS_i.

$$LBS_i = LBS + \sum_{r_i=r_{i-1}}^{r_{i-1}+N} V_{i+1}(r_i\ ,\ t_{i-1}+N)f_N(r_i) \qquad (3.44)$$

Where LBS is given by

$$LBS = \text{Min}\left\{K(N)\ ,\ K(N) + (N-1)(F(N-1)-k_v+k_r(1-r_{i-1}/t_{i-1}))\right\}$$

Clearly, by using Theorems 3.7 and 3.8; LBS is substituted by $K(n=1,c=-1)$ when $r_{i-1}/t_{i-1} > \text{Max}\left\{p_c(1+N/t_{i-1}) \ , \ 1-k_v/k_r\right\}$ and by $K(n=1,c=1)$ when

$$r_{i-1}/t_{i-1} < \text{Min}\left\{N/t_{i-1}(p_c-1) + p_c \ , \ k_v/(k_a-k_r)\right\}$$

The property that the second term of S_i is non-increasing in n_i provides a lower bound for n_i^+; $i=1,\ldots,l-1$. Notice that S_i evaluated at n_i^* can only be improved if we increase the sample size. That is $n_i^* \leq n_i^+$ for $i=1,\ldots,l-1$.

When considering a sequence of lots, an increase of the size of the sample of a lot reduce the losses from wrong decisions in that lot. In addition, as we use this sampling information in our subsequent lots it also reduces the losses from wrong decisions in subsequent lots. Therefore, the optimal sample size of lot i in a given state, say (r,t), is greater or equal to the optimal sample size of a single lot at (r,t).

3.4.4 General Results

In this section we develop some results which will simplify the fundamental recursion relation (3.43).

Theorem 3.9.

$$\sum_{r_{i-1}=r}^{r+n_i^+} V_i(r_{i-1},t+n_i^+)g_{n_i^+}(r_{i-1}) \leq \sum_{r_i=r}^{r+n_i^+} V_{i+1}(r_i,t+n_i^+)g_{n_i^+}(r_i) + \text{Min}\left\{k_r \ , \ k_a r/t\right\}$$

for $i=1,2,..,l$.

Proof:

By backward induction

for i=1

$$\sum_{r_{1-1}=r}^{r+n_2^+} V_1(r_{1-1},\ t+n_1^+)g_{n_2^+}(r_{1-1}) = \sum_{x_{1-1}=0}^{n_2^+} V_1(r+x_{1-1},\ t+n_1^+)g_{n_2^+}(x_{1-1})$$

$$\leqslant \text{Min}\left\{k_r,\ k_a r/(t+n_1^+)\right\} g_{n_2^+}(0) + \text{Min}\left\{k_r,\ k_a(r+1)/(t+n_1^+)\right\} g_{n_2^+}(1)$$
$$+ \cdots\cdots\cdots + \text{Min}\left\{k_r,\ k_a(r+n_1^+)/(t+n_1^+)\right\} g_{n_2^+}(n_1^+)$$

$$\leqslant \text{Min}\left\{k_r\ ,\ k_a r/t\right\}$$

Last inequality follows as

$$k_a/(t+n_1^+) \sum_{x_{2-1}=0}^{n_2^+} (r+x_{1-1})g_{n_2^+}(x_{1-1}) = k_a r/t$$

Assume true for i. For i-1 we have

$$V_{i-1}(r,t+n_{i-1}^+) \leqslant \text{Min}\begin{bmatrix} k_r+V_i(r,t+n_{i-1}^+) \\ k_a r/(t+n_{i-1}^+) + V_i(r,t+n_{i-1}^+) \end{bmatrix} = V_i(r,t+n_{i-1}^+) + \text{Min}\left(k_r, k_a r/(t+n_{i-1}^+)\right)$$

$$V_{i-1}(r+1,t+n_{i-1}^+) \leqslant \text{Min}\begin{bmatrix} k_r+V_i(r+1,t+n_{i-1}^+) \\ k_a(r+1)/(t+n_{i-1}^+) + V_i(r+1,t+n_{i-1}^+) \end{bmatrix} =$$
$$V_i(r+1,t+n_{i-1}^+) + \text{Min}\left\{k_r\ ,\ k_a(r+1)/(t+n_{i-1}^+)\right\}$$

$$\vdots$$

$$V_{i-1}(r+n_{i-1}^+\ ,\ t+n_{i-1}^+) \leqslant \text{Min}\begin{bmatrix} k_r+V_i(r+n_{i-1}^+,\ t+n_{i-1}^+) \\ k_a(r+n_{i-1}^+)/(t+n_{i-1}^+) + V_i(r+n_{i-1}^+,t+n_{i-1}^+) \end{bmatrix} =$$
$$V_i(r+n_{i-1}^+,t+n_{i-1}^+) + \text{Min}\left\{k_r\ ,\ k_a(r+n_{i-1}^+)/(t+n_{i-1}^+)\right\}$$

therefore,

$$\sum_{r_{i-2}=r}^{r+n^+_{i-1}} V_{i-1}(r_{i-2}\ ,\ t+n^+_{i-1})\, g_{n^+_{i-1}}(r_{i-2}) \leqslant \sum_{r_{i-1}=r}^{r+n^+_{i-1}} V_i(r_{i-1},\ t+n^+_{i-1})\, g_{n^+_{i-1}}(r_{i-1})$$
$$+ \sum_{x_{i-1}=0}^{n^+_{i-2}} \mathrm{Min}\left\{k_r\ ,\ k_a(r+x_{i-1})/(t+n^+_{i-1})\right\} g_{n^+_{i-1}}(x_{i-1})$$
$$\leqslant \sum_{r_{i-1}=r}^{r+n^+_{i-1}} V_i(r_{i-1},\ t+n^+_{i-1})\, g_{n^+_{i-1}}(r_{i-1}) + \mathrm{Min}\left\{k_r\ ,\ k_a r/t\right\} \quad \blacksquare$$

Theorem 3.10

If sampling is optimal at state (r,t) of stage i, then sampling is optimal at state (r,t) of stage i-1; i=2,...,1.

Proof:

For i=2,...,1

$$V_i(r,t) = K'(n_i^+) + \sum_{r_i=r}^{r+n_i^+} V_{i+1}(r_i\ ,\ t+n_i^+)g_{n_i^+}(r_i) \quad n_i^+ \in (1,N)$$

$$V_{i-1}(r,t) = \text{Min}\begin{bmatrix} S_{i-1}: K'(n_{i-1}^+) + \sum_{r_{i-1}=r}^{r+n_{i-1}^+} V_i(r_{i-1},\ t+n_{i-1}^+)g_{n_{i-1}^+}(r_{i-1}) \\ R_{i-1}: k_r + K'(n_i^+) + \sum_{r_i=r}^{r+n_i^+} V_{i+1}(r_i\ ,\ t+n_i^+)g_{n_i^+}(r_i) \\ A_{i-1}: k_a r/t + K'(n_i^+) + \sum_{r_i=r}^{r+n_i^+} V_{i+1}(r_i\ ,\ t+n_i^+)g_{n_i^+}(r_i) \end{bmatrix}$$

where $n_i^+ \in (1,N)$ and $n_{i-1}^+ \in (1,N)$

To prove that $S_{i-1} \leq \min\{R_{i-1},\ A_{i-1}\}$ we only need S_{i-1} evaluated at n_i^+ to be less than or equal to $\min\{R_{i-1},\ A_{i-1}\}$ i.e.,

$$\sum_{r_{i-1}=r}^{r+n_i^+} V_i(r_{i-1},\ t+n_i^+)g_{n_i^+}(r_{i-1}) \leq \sum_{r_i=r}^{r+n_i^+} V_{i+1}(r_i,\ t+n_i^+)g_{n_i^+}(r_i) + \text{Min}(k_r, k_a r/t)$$

for i=2,...,1

Which follows by Theorem 3.9.

Corollary 3.1

If sampling is optimal at state (r,t) of stage h, $2 \leq h \leq 1$; then sampling is optimal at state (r,t) of stages 1,2,..., h-1.

Proof:

Sampling optimal at state (r,t) of stage h implies by Theorem 3.10, that sampling is optimal at state (r,t) of stage h-1. By repetition of the same argument result follows. ■

Corollary 3.2

If a non-sampling action is optimal at state (r,t) of stage h, $1 \leq h \leq l$; then the same action is optimal at stages h+1,...,l.

Proof:

Assume reject is optimal at state (r,t) of stage h, then,

$$V_h(r,t) = k_r + V_{h+1}(r,t) \leqslant k_a r/t + V_{h+1}(r,t)$$

therefore

$$k_r \leqslant k_a r/t$$

By Corollary 3.1, sampling is not an optimal policy at state (r,t) of stages h+1,...,l. Therefore

$$V_j(r,t) = \text{Min} \begin{bmatrix} k_r + V_{j+1}(r,t) \\ k_a r/t + V_{j+1}(r,t) \end{bmatrix} = k_r + V_{j+1}(r,t) \quad \text{for } j=h+1,\ldots,l$$

i.e., reject is optimal for stages h+1,...,l.

Using the same argument we can prove that if accept is optimal at state (r,t) of stage h, $1 \leq h < l$, then accept is optimal at stages h+1,...,l. ■

It is important to notice that the proof of Theorem 3.9 does not depend on the form of K(n); and therefore results given in Theorem 3.10 and Corollaries 3.1 and 3.2 are independent of the model used to

describe the expected cost of sampling and, clearly, of the type of prior distributions assumed.

Corollary 3.2 tells us that if a non-sampling action is optimal at stage i then the same action is optimal for the rest of the lots. For example, if $V_i(r,t) = k_r + V_{i+1}(r,t)$, then $V_{i+1}(r,t) = k_r + V_{i+2}(r,t)$, $V_{i+2}(r,t) = k_r + V_{i+3}(r,t),\ldots,V_l(r,t) = k_r$.

Therefore, the recursive relation (3.43) is equivalent to

$$V_i(r_{i-1},t_{i-1}) = \text{Min}\begin{bmatrix} S_i:\ K(n_i^+) + \sum_{r_i=r_{i-1}}^{r_{i-1}+n_i^+} V_{i+1}(r_i,t_{i-1}+n_i^+)g_{n_i^+}(r_i) \\ R_i:\ (l-i+1)k_r \\ A_i:\ (l-i+1)k_a r_{i-1}/t_{i-1} \end{bmatrix} \qquad (3.45)$$

where $n_i^+ \in (1,N)$ for $i=1,2,..,l$ and

with boundary conditions $V_{l+1}(.,.) = 0$.

In (3.45) the reject and accept alternatives are no longer a function of $V_{i+1}(r_{i-1},t_{i-1})$.

The number of states that we have to consider to find the exact solution is slightly decreased. Let's assume the same example used in Section 3.4.1, that is, a sequence of 3 lots of size 2 each, with the first lot at state (1,10). Therefore, at stage 1 we only consider state (1,10); at stage 2 we need to consider states (1,11) and (2,11) if $n_1=1$, and states (1,12), (2,12) and (3,12) if $n_1=2$; at stage 3, states (1,12), (2,12) and (3,12) if $n_1=1$ and $n_2=1$, states (1,13), (2,13), (3,13) and (4,13) if $n_1=1$ and $n_2=2$ or if $n_1=2$ and $n_2=1$, and states (1,14), (2,14), (3,14), (4,14) and (5,14) if $n_1=2$ and $n_2=2$.

Therefore, we need to consider one state at stage 1, five states at stage 2, and twelve states at stage 3, compared to 1, 6 and 15 possible states at stages 1, 2 and 3, respectively.

In general, if we define:

NS'_i = Number of states to consider in stage i ; i=1,2,..,l

then $NS'_i = (L_i+1)(L_i+2)/2 - i(i-1)/2$; i=1,2,..l (3.46)

where: $L_i = (i-1)\ N$; i=1,...,l

Now if we have a sequence of five lots of 1000 items each, then $L_5 = 4000$ and $NS'_5 = 8,005,991$.

3.4.5 Convergence of Optimal Policy for Lot 1

If the process quality, p, is known, the prior distribution of the lot quality, $f_N(X)$, given by Equation (3.5) becomes binomial with parametes N and p, b(X,N,p). By the reproducibility of the binomial prior distribution, Hald (7), $g_n(x) \sim b(x,n,p)$ and $P(y/x) \sim b(y,N-n,p)$. Therefore, $E(x) = np$ and $E(p^*/x) = E(y/x)/(N-n) = p$, which agrees with Mood (16).

The expected cost of sampling, Equation (2.8) becomes

$$K(n,c) = k_I + k_v n + k_r(np+N-n) + (N-n)(k_a p - k_r)\sum_{x=0}^{c} g_n(x) \qquad (3.47)$$

It is clear that the optimal value of c for fixed n, c_n, is given by

$$c_n = \begin{cases} n & ; \text{ if } p < p_c \\ -1 & ; \text{ otherwise} \end{cases}$$

Therefore K(n); n=1,...,N is given by

$$K(n)=\begin{cases} k_I + k_v n + k_r np + (N-n)k_a p & ; \text{ if } p < p_c \ ; \ c_n = n \\ k_I + k_v n + k_r (np+N-n) & ; \text{ otherwise } ; \ c_n = -1 \end{cases} \qquad (3.48)$$

Note that Equation (3.48) can be obtained by Equations (2.11) and (2.12) with $E(x) = np$ and $E(X) = Np$.

Theorem 3.11

For a process with quality p, the optimal policy for lot 1 when $1 \to \infty$ is:

$$\text{If } E_1(p) < p_c \quad \text{then} \begin{cases} \text{Accept without sampling} & ; \text{ if } k_I/N + k_v > E_1(p)(k_a-k_r) \\ \text{100\% Sampling} & ; \text{ otherwise} \end{cases}$$

$$\text{If } E_1(p) \geq p_c \quad \text{then} \begin{cases} \text{Reject without sampling} & ; \text{ if } k_I/N + k_v \geq k_r(1-E_1(p)) \\ \text{100\% Sampling} & ; \text{ otherwise} \end{cases}$$

Proof:

Assume we know the process quality, p. Then $A_1 = k_a p$ and S_1 is given by Equation (3.48).

a) If $p < p_c$, then $A_1 < R_1$.

$A_1 < S_1 = (k_I + k_v n + k_r np + (N-n)k_a p)/N$ for $n=1,\ldots,N$

If and only if, $k_v > p\,(k_a-k_r) - k_I/n$ For $n=1,\ldots,N$.

It is clear that this last condition is true if and only if,

$$k_I/N + k_v > p(k_a-k_r)$$

If $A_1 \geq S_1$ Then by Theorem 3.6 100% sampling is optimal.

b) If $p \geq p_c$, then $R_1 \leq A_1$.

$R_1 \leq S_1 = (k_I + k_v n + k_r\,(np + N-n))/N$ for $n=1,\ldots,N$

If and only if, $k_v \geq k_r\,(1-p) - k_I/n$ for $n=1,\ldots,N$.

This last condition is true if and only if $k_I/N + k_v \geq k_r(1-p)$.

If $R_1 > S_1$ then by Theorem 3.6 100% sampling is optimal.

Recall $E_1(p) = (r_0 + \sum_{j=1}^{l-1} x_j)/(t_0 + \sum_{j=1}^{l-1} n_j)$;

By letting $$n' = \sum_{j=1}^{l-1} n_j$$

Then $$E_1(p) = \frac{r_0 + \sum_{j=1}^{l-1} x_j}{t_0 + n'} = \frac{r_0/n' + \sum_{j=1}^{l-1} x_j/n'}{t_0/n' + 1}$$

Note that as $l \to \infty$, $n' \to \infty$ and $\sum_{j=1}^{l-1} x_j/n' \to p$

Therefore $E_1(p) \to p$ as $l \to \infty$. ■

Theorem 3.11 tells us that for $l \to \infty$, the optimal policy for the last lot tends to be either a non-sampling alternative or 100% sampling. This is, of course, a result of the assumption that the process is operating in a random manner with quality p. No shifts in the process quality are allowed. In practice, it is widely known that any process will tend to go out of control and therefore shifts in the process quality are expected.

CHAPTER 4

SOLUTIONS

4.1 An Algorithm for the Optimal Sampling Plan of a Single Lot

The algorithm proposed is based on the one presented by Moskowitz and Berry (18) with the use of a more efficient search thechnique.

Moskowitz and Berry define switchover points as the values of n where the value of c_n changes. In Figure 2, the switchover points are the dotted points in the graph.

They presented two main results which were, however, only checked experimentally. Firstly, they stated that there exists only one point such that $K(n_a) > K(n_b) < K(n_c)$

where $n_a < n_b < n_c$ and $c_{n_a} < c_{n_b} < c_{n_c}$

and that the optimal sample size lies somewhere between the switchover points n_a and n_c.

As discussed in Section 3.3.1, the maximum increase of c_n when n is increased by one is one. Therefore, we may use $c_{n_a} = c_{n_b} + 1$ and $c_{n_b} = c_{n_c} + 1$.

Define a ditonic function as a discrete function g(j) such that g(j) is decreasing for $1 \leq j \leq j^*$ and increasing for $j > j^*$. That is, $\Delta_j\, g(j) < 0$ for $1 \leq j \leq j^*$ and $\Delta_j\, g(j) > 0$ for $j > j^*$.

By using Moskowitz and Berry's results we may conclude that K(.) evaluated at the switchover points is a ditonic function. They also

proposed that K(n), evaluated between contiguous switchover points, that is for c_n constant, is also a ditonic function. Both results have been supported by a wide variety of experimental values in our research.

The minimum of a ditonic function g(j) j=1,...,NP can efficiently be found by using the following search procedure.

The sequence of Fibonacci numbers is 1,1,2,3,5,8,13,21,..... . That is, each number is the sum of the previous two numbers. Let F_i = the i+1 number of the Fibonacci sequence. The procedure can be described in the following steps.

a) Find the smallest Fibonacci number bigger than NP, F_i. If i=3 then calculate g(1) and g(2) to find the minimum and stop. If i>3 calculate $g(F_{i-1})$ and go to next step.

b) Calculate $g(F_{i-2})$; if $g(F_{i-1}) \leq g(F_{i-2})$ set $KP=\min\{2F_{i-2},NP\}$ and calculate g(KP), if $g(KP) \leq g(F_{i-1})$ evaluate function in interval $F_{i-1}+1$ to min (F_i,NP), evaluate function in interval $F_{i-2}+1$ to KP otherwise; procedure terminates.

If $g(F_{i-1}) > g(F_{i-2})$ set i=i-1, if i=2 stop, repeat step b) otherwise.

If we let n(1), n(2),.... be the switchover points, the algorithm for finding the optimal sampling plan of a single lot can be divided in the following three steps.

1) Find all switchover points n(1), n(2),... in the interval (1,N). By using the proposed search find the switchover point that minimizes the expected cost of sampling, which will be denoted by n(s).

2) Define $L = \{n:n(s-1) < n < n(s)\}$ and $U = \{n:n(s) < n < n(s+1)\}$. By using the proposed search find the $n \in L$ that minimizes $K(n)$, n_L; and the $n \in U$ that minimizes $K(n)$, n_U.

3) The optimal expected cost and consequently the optimal acceptance plan (n^*, c_n) is such that $K(n^*) = \text{Min}\{K(n(s)), K(n_L), K(n_U)\}$.

4.2 Exact Solution

The exact solution to our problem is given by the recursive relation (3.45). To find $V_i(.,.)$ for a given state we have first to find the n_i which minimizes S_i, n_i^+; and then compute $\text{Min}\{S_i, R_i, A_i\}$.

However, to find the optimal policy to our problem we need to evaluate relation (3.45) for every state of every stage. As discussed in Section 3.4.4, the number of states to consider at a given stage increases rapidly with the lot size and the number of the stage. Therefore, a complete enumeration becomes cumbersome and inefficient. The proposed method uses all of the features described in the previous chapters.

First we discuss the single lot case, which is a special case of our more general model with $l=1$. Note that for this case relation (3.45) becomes

$$V_1(r_o,t_o) = \text{Min}\{K'(n^*), k_r, k_a r_o/t_o\}$$

To find $V_1(r_o,t_o)$ we first use Theorem 3.5 to find out whether 100% sampling is optimal or not. If 100% sampling is optimal then

$V_1(r_o,t_o) = \text{Min}\{ k_I/N + k_v + k_r\, r_o/t_o, k_r, k_a\, r_o/t_o\}$. Otherwise, we calculate the LBS by using Equation (3.34) and then test whether the $\text{Min}\{\text{LBS}, k_r, k_a\, r_o/t_o\} < \text{LBS}$ or not. If true then $V_1(r_o, t_o) = \text{Min}\{k_r, k_a\, r_o/t_o\}$, if false then we use the algorithm discussed in Section 4.1 to find $K'(n^*)$ and compute $V_1(r_o,t_o) = \text{Min}\{K'(n^*), k_r, k_a\, r_o/t_o\}$.

For $l > 1$, the lower bound for n_i^+ discussed in Section 3.4.3 can reduce further the number of states to be considered at stages $2,3,\ldots,l$. If $n_1^* > 1$ and as $n_1^+ \geq n_1^*$, then the number of states that we have to consider in stages $2,\ldots l$ is smaller than given in Equation (3.46). For example, assume we have a sequence of two lots of five items each ($l=2$ and $N=5$) with lot 1 at state (1,10). Further, let's assume $n_1^* = 3$. Then the states to consider at stage 2 are (1,13), (2,13), (3,13), and (4,13); (1,14), (2,14),...,(5,14); and (1,15), (2,15),...,(6,15). That is 15 states, compared to $NS_2 = 20$ given by Equation (3.46). Clearly, the reduction of states will depend on n_1^*, the bigger n_1^* the more reduction.

No further reduction of states is considered by using the lower bound $n_i^+ \geq n_i^*$. This is because we would need to find the optimal sample size for single lots at every state in stage 2 and so on. In our example, stage 2 would require finding the optimal sample size of a single lot at the 15 states to consider, this is clearly computationally inefficient compared to the possible savings.

The procedure for finding the optimal policy can be summarized in the following steps.

1) Find n_1^*. To do this we use Theorem 3.5 to find out whether 100% sampling is optimal or not. If 100% sampling is optimal then $n_1^* = N$. Otherwise, we find n_1^* by the search procedure. At this point the states to consider at stages 2,...,1 are defined.

2) Solve stage 1. Here we first use Theorems 3.7 and 3.8 to find all states which can be solved immediately. States which do not satisfy conditions are then tested by Theorem 3.5 to find out whether 100% sampling is optimal or not, and construct the lower bound of sampling, LBS. The LBS is stored for subsequent use. If 100% sampling is optimal, $V_1(.,.)$ is calculated immediately. Otherwise, we test whether the Min $\{R_1, A_1, LBS\} < LBS$ or not. If true, then $V_1(.,.) = \text{Min}\{R_1, A_1\}$. If false then we use the search procedure of Section 4.1 to find $K(n^*)$ which is S_1, and compute $V_1(.,.) = \text{Min}\{S_1, R_1, A_1\}$.

3) Solve stages i=1,...,1-1. Starting with stage 1-1 we do the following for every stage.

For every stage considered, we first calculate the lower bound for S_i, LBS_i, described by Equation (3.44). Recall that the first term of LBS_i, LBS, is replaced by $K(n^*)$ when the conditions of Theorems 3.7 or 3.8 are satisfied. In addition, by using the property discussed in Section 3.4.3; for all states that at stage 1 100% sampling is the optimal acceptance plan, we have $LBS_i = S_i$ and therefore $V_i(.,.)$ can be calculated immediately by $V_i(.,.) = \text{Min}\{LBS_i, R_i, A_i\}$. For the rest of the states we test whether the Min $\{LBS_i, R_i, A_i\} < LBS_i$ or not. If true, then $V_i(.,.) = \text{Min}\{R_i, A_i\}$.

If false then we use complete ennumeration to find S_i and then compute $V_i(.,.) = \text{Min}\{S_i, R_i, A_i\}$.

4.3 Approximate Solutions

As we have discussed earlier, industrial size problems will usually be too big for the exact method, therefore in this section we shall discuss approximate solutions.

The reason for the magnitude of the problem is the number of states that a lot can assume. The principal reason for having such an explosion of states is that the sample size of any lot can vary from one to the lot size. Therefore, in order to simplify the problem, we may only allow one sample size for lots 1,...,l-1. This measure reduces drastically the number of states to consider at all stages. For example, if we consider a sequence of five lots of 1000 items each, and we fix $n_1 = n_2 = \ldots = n_5 = 50$; then the states to consider at stage 5 are $(r_0, t_0 + 200)$, (r_0+1, t_0+200),..., (r_0+200, t_0+200). That is 201 instead of $NS'_5 = 8{,}005{,}991$. Here we are not taking into account the possible reduction in states when we use the lower bound $n_1^+ \geq n_1^*$ discussed in Section 4.2.

This drastic reduction is a result of Corollary 3.2. For example, we do not need to consider state (r_0, t_0+100) at stage 5 because the only way that an optimal policy is going to take us to that state in stage 5 is by sampling lot 1 ($n_1 = 50$), sampling lot 2 ($n_2 = 50$) and then using a non-sampling action for lots 3 and 4. Therefore, we can evaluate this policy in stage 3 instead.

In general, the number of states to consider in this case at stage i, NSA_i, is:

$$NSA_i = \sum_{j=1}^{i-1} n_j + 1 \quad i=1,2,..,l \tag{4.1}$$

In order to have a yardstick for comparison we propose two different methods for fixing the sample sizes of each lot.

As we discussed in Section 3.4.3, n_1^* is a lower bound for n_1^+, therefore we would like to fix n_1 such that $n_1 \geq n_1^*$. Furthermore, whenever the optimal policy for a sequence of l lots is reject or accept all lots without sampling or sample lot 1 followed by rejection or accepting the rest of the lots without sampling for every possible outcome of the sample performed in lot 1; the solution for a sequence of l lots is identical to the solution of a single lot with lot size lN. As an example, assume l=3 and N=10 with $(r_0, t_0) = (1,10)$. Further assume the optimal policy is: sample lot 1 with $n_1^+=2$; if $x_1=0$, then accept without sampling lots 2 and 3; and if $x_1=1$ or $x_1=2$, then reject without sampling lots 2 and 3. Then the same solution can be found by solving the single lot case l=1 and N=30 with $(r_0, t_0) = (1,10)$.

This phenomenon is due to the fact that S_i is the expected cost of sampling for the sequence, and if only non-sampling alternatives are involved in $V_{i+1}(.,.)$ for i=1,...,l-1; S_1/l in relation (3.45) is equal to the expected cost of sampling for a lot of size lN. In addition, from relation (3.45) A_1/l and R_1/l then become k_r and k_n r_0/t_0, respectively.

This property is the basis for Method 1. Both methods can be classified as a forward-backward dynamic programming procedures, where the forward pass finds the sample sizes allowed for each lot $n_1, n_2,\ldots,n_{l-1} > 0$; and the backward pass uses relation (3.45) to find the optimal policy for the reduced state space. An upper limit for the sample sizes is used in those cases where large sample sizes are not desired, and to avoid the possibility of sample sizes greater than the lot size.

Method 1.

Forward Pass. If we let

$n^*(N,r,t)$ = optimal sample size of a single lot at state (r,t) with lot size N, and

UL = upper limit; $1 \leq UL \leq N$

We then set $$n_1 = \begin{cases} n^*(lN,r_0,t_0) & ; \text{ if } n^*(lN,r_0,t_0) \leq UL \\ UL & ; \text{ otherwise} \end{cases}$$

From relation (3.45) the only possible states for lot 2 given that we sampled lot 1 are $(r_0,\ t_0+n_1)$, $(r_0+1,\ t_0+n_1)$,....., $(r_0+n_1,\ t_0+n_1)$. Therefore, the a priori best guess of the state of lot 2 given that we sampled lot 1 is $(r_0+E(x_1),\ t_0+n_1)$, where $E(x_1) = n_1 r_0/t_0$ we then set

$$n_2 = \begin{cases} n^*((l-1)N,r_0+E(x_1),t_0+n_1); \text{if } n^*((l-1)N,r_0+E(x_1),t_0+n_1) \leq UL \\ UL \qquad\qquad ; \text{ otherwise} \end{cases}$$

The only possible states of lot 3 given that we sampled lots 1 and 2 are $(r_0, t_0+n_1+n_2)$, $(r_0+1, t_0+n_1+n_2)$,....,$(r_0+n_1+n_2, t_0+n_1+n_2)$. Recall that non-sampling for lot 1 and then sampling lot 2 cannot be an optimal policy by Corollary 3.2. Therefore, the a priori best guess on the state of lot 3 given that we sampled lots 1 and 2 is $(r_0 + E(x_1 + x_2), t_0 + n_1 + n_2)$, where $E(x_1 + x_2) = (n_1 + n_2)\, r_0/t_0$. We set n_3 similarly to n_1 and n_2. In general we define

$$n_i = \begin{cases} n^*((1-i+1)N, r_0+E(x_1+\ldots+x_{i-1}), t_0+n_1+\ldots+n_{i-1}) & ; \text{ if } n^*(.,.,.) \leq UL \\ UL & ; \text{ otherwise} \end{cases}$$

For $i=1,\ldots, 1-1$, where $E(\sum_{j=1}^{i-1} x_j) = r_0 \sum_{j=1}^{i-1} n_j/t_0$

The forward pass terminates when n_{1-1} is calculated.

Backward Pass.

It is important to recall first that in this case we have t fixed at every stage and r varying from r_0 to $r_0 + \sum_{j=1}^{i-1} n_j$ in stages $i=2,\ldots,1$. The backward pass is the solution to relation (3.45) with n_i fixed for $i=1,\ldots,1-1$. It can be divided into the following parts. Part 1) Solve stage 1. First we use Theorems 3.7 and 3.8 to find all the states that can be solved immediately. States which do not satisfy the conditions are then tested by Theorem 3.5 to find out whether 100% sampling is optimal or not and computing LBS. If 100% sampling is optimal, $V_1(., t_0 + \sum_{j=1}^{1-1} n_j)$ is calculated immediately. Otherwise, we test whether or not the $\text{Min}\{R_1, A_1, LBS\} < LBS$ or not. If true then $V_1(., t_0 + \sum_{j=1}^{1-1} n_j) = \text{Min}\{R_1, A_1\}$, if false then we use the search procedure to find S_1 and compute

$$V_1(\cdot,\ t_0+\sum_{i=1}^{t-1} n_j) = \text{Min}\ \{S_1, R_1, A_1\}$$

Part 2) Solve stages i=1,...,1-1.

By Theorem 3.7 it is optimal to reject without sampling a single lot at state (r,t) if:

$$r/t = E(p) > \text{Max}\ \{p_c(1+N/t),\ 1-k_v/k_r\} \qquad (4.2)$$

It is easy to see that if (4.2) is satisfied, then a single lot at state (r+x,t), where $x \geq 0$, has as the optimal policy reject without sampling. Further, if (4.2) holds, then rejecting without sampling is optimal for a single lot at state (r,t-n), where $n \geq 0$. This last statement can be proved as follows

$$r/t > p_c(1+N/t) \quad \text{implies} \quad r > p_c(t+N) > p_c(t-n+N)$$

therefore $\quad r > p_c(t-n+N) = p_c(t-n)(1 + N/(t-n))$

and $\quad r/(t-n) > p_c(1 + N/(t-n))$

which implies $\quad r/(t-n) > \text{Max}\ \{p_c(1 + N/(t-n)),\ 1-k_v/k_r\}$

As soon as condition (4.2) is satisfied at stage 1, then we know that $V_1(r+x,t) = k_r$ for x=1,2,... . And as we have fixed the sample size to take at stage 1-1, n_{1-1}, then

$$V_{1-1}(r,t-n_{1-1}) = \text{Min} \begin{bmatrix} K'(n_{1-1}) + k_r \\ 2k_r \\ 2k_a r/(t-n_{1-1}) \end{bmatrix}$$

In addition, by the proof above, it is optimal to reject a single lot at $(r,t-n_{1-1})$. Therefore,

$\text{Min}\left\{K'(n_{l-1}) \ , \ k_r \ , \ k_a r/(t-n_{l-1})\right\} = k_r$; and $V_{l-1}(r,t-n_{l-1}) = 2k_r$

Using the same argument, we can prove that if (4.2) holds at stage 1 and only one sample size is allowed for each lot, then

$V_{l-i}(r+x,\ t-n_{l-1}-\cdots-n_{l-i}) = (i+1)k_r$ for $x \geq 0$ and $i=1,2,..,l-1$

That is, once we find the r_{l-1} at stage 1 such that condition (4.2) is satisfied, then all states $(r_{i-1},\ t_{i-1})$ for stages $i=1,...,l-1$ with $r_{i-1} \geq r_{l-1}$ have as the optimal policy reject without sampling.

By using this result, the proposed procedure to solve for the optimal sampling policy at stages $i=1,..,l-1$ is as follows:

Starting with stage $l-1$ and moving backwards do the following for every stage i.

a) Set $r_{i-1} = r_o$

b) Test whether $r_{i-1} \geq r_{l-1}$ or not. If true then set

$$V_i\ (r, t_o + \sum_{j=1}^{i-1} n_j) = (i+1)\ k_r \text{ for } r = r_{i-1}, \ldots, r_o + \sum_{j=1}^{i-1} n_j,$$

Set $i=i-1$ and go to step a).

If false then compute

$$S_i = K'(n_i) + \sum_{x_i=0}^{n_i} V_{i+1}(r_{i-1}+x_i \ , \ t_o + \sum_{j=1}^{i} n_j) g_{n_i}(x_i)$$

and $V_i\ (r_{i-1},\ t_o + \sum_{j=1}^{i-1} n_j) = \text{Min}\left\{S_i,\ R_i,\ A_i\right\}$

Then if $r_{i-1} < r_o + \sum_{j=1}^{i-1} n_j$, set $r_{i-1} = r_{i-1} + 1$ and repeat step b). Otherwise set $i=i-1$ and go to step a).

Method 2.

This method is a simplified version of Method 1. Again we fix the sample size of each lot in order to reduce the possible number of

states. Two different versions of this method were tested. We will call them Method 2A and Method 2B. In the forward pass of Method 2A we set

$$n_1=n_2=\cdots=n_{l-1}= \text{Min}\left\{n^*(lN,r_o,t_o)\ ,\ UL\right\}$$

And in Method 2B we use

$$n_1=n_2=\cdots=n_{l-1}= \text{Min}\left\{n^*(N,r_o,t_o)\ ,\ UL\right\}$$

The backward pass for both methods is the same as in Method 1.

4.4 Programs Organization

The exact method and approximate methods described in Sections 4.2 and 4.3 have various common procedures, therefore three subroutines were developed for the use of the four methods. The main programs and subroutines were written in Fortran IV. Program listings and flowcharts are included in the Appendix.

SUBROUTINE FUNC (SS,F,C,TC).

The parameters in terms of our notation are: SS=n, F=F(n), C=c_n and TC=K(n).

This subroutine calculates first c_n by using Equations (3.20) and (3.20a). It then calculates F(n) by Equation (3.22) if $0 \leqslant c_n < n$; by $F(n) = k_a E(P) - k_r$ if $c_n=n$; and sets F(n)=0 if $c_n = -1$ (Equation (2.12)). Finally calculates K(n) by using Equation (3.21).

SUBROUTINE SEARCH.

This subroutine is the algorithm for searching for the optimal sampling plan described in Section 4.1. Before starting the

algorithm and in order to try to reduce the possible values that the sample size can take; subroutine search calculates first the upper bound for n, given by Equation (3.39), and then finds the largest n with c_n=n, or c_n = -1, if any.

SUBROUTINE FIBON (PP, NP, IS, SST, CST, TCST).

The parameters are defined as:

PP = Vector which elements are the sample sizes to search from

NP = Number of elements in PP (NP $\geq$ 2)

IS = The element number of the sample size which attains the minimum expected cost

SST = The minimum cost sample size, where the sample size is an element of PP

CST = The minimum cost acceptance number

TCST = The minimum expected cost of sampling

This subroutine uses the search procedure for ditonic functions described in section 4.1. The vector PP contains the set of sample sizes to search from. For example, if the switchover points are the sample sizes 1,9,18,27,36 and 45 then K(PP(3)) = K(18) , K(PP(5)) = K(36) and so on.

A detailed flow chart of the subroutine is given in the appendix.

CHAPTER 5

RESULTS AND CONCLUSIONS

5.1 Test Results

To test the effectiveness of the lowerbounds discussed in Sections 3.3.2 and 3.4.3 and the results of Theorems 3.5, 3.7 and 3.8; three examples with the following data were solved by using the exact method.

Table 2.

Example	l	k_I	k_v	k_r	k_a	r_o	t_o	N	$E(p)$	p_c
1	3	0	0.6	0.7	14	1	21	30	0.0476	0.05
2	3	0	1.1	1.0	15	1	12	30	0.0833	0.0667
3	3	0	1.1	1.0	15	1	21	30	0.0476	0.0667

Data for examples 1-3

Note that the parameters were chosen so that $E(p)$ is close to p_c in Example 1. $E(p)$ is greater than p_c in Example 2 and $E(p)$ is less than p_c in Example 3.

The following table shows the number of states where the algorithm had to search for the optimal sample size versus the number of possible states at each stage (NS_i) for all three examples.

Table 3

		Stage i		
		1	2	3
Number of possible states= NS_i		1	496	1891
Number of states where n_i^+ had to be searched	Ex. 1	1 (100%)	86 (17.3%)	82 (4.3%)
	Ex. 2	1 (100%)	71 (14.3%)	144 (7.6%)
	Ex. 3	1 (100%)	94 (18.6%)	166 (8.8%)

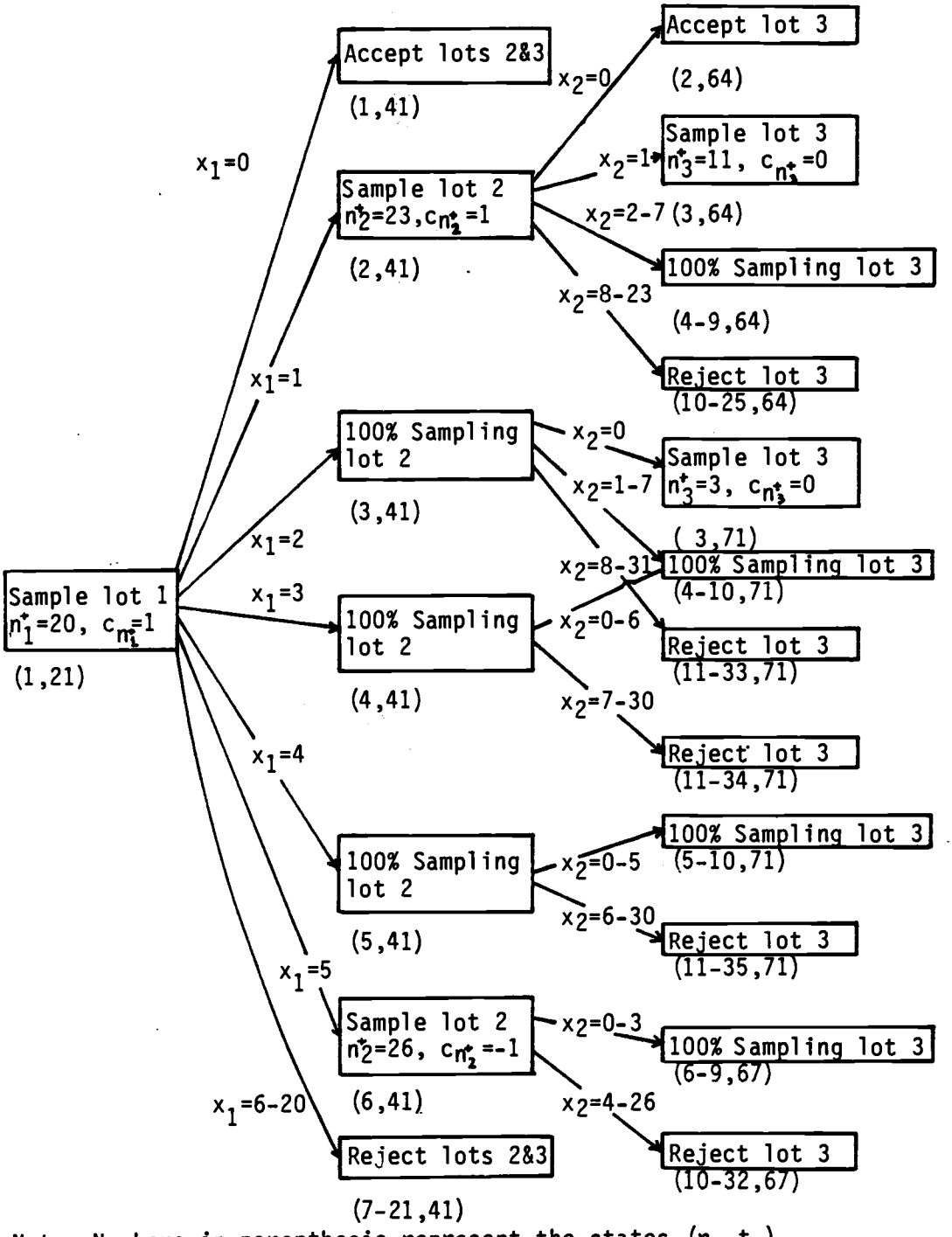
Number of possible states and number of states where the algorithm had to search for the optimal sample size

In Example 2, the optimal sample size was searched in 144 states out of 1891 possible states at stage 3 (7.6%). 1747 states were pruned by either the lower bound LBS or by one of the theorems 3.5, 3.7 or 3.8.

The optimal policy for Example 1 is described in Figure 3.

We start with lot 1 at state $(r_0,t_0) = (1,21)$. The optimal plan for lot 1 is $(n_1^+, c_{n_1^+}) = (20,1)$. Let's assume we take a sample of 20 items and we find one defect, i.e. $x_1 = 1$. Therefore, we accept Lot 1 and the prior distribution of Lot 2 updates to $(r_1, t_1) = (2,41)$. The optimal plan for Lot 2 given that $x_1 = 1$ is then $(n_2^+, c_{n_2^+}) = (23,1)$. Let's assume we take the sample of 23 items from Lot 2 and we find two defects, i.e. $x_2 = 2$. Therefore, we reject Lot 2 and the prior distribution of Lot 3 updates its parameters to $(r_2, t_2) = (4,64)$. The optimal policy for Lot 3 given the results for Lots 1 and 2 is to 100% sample the lot, that is, total inspection.

The approximate methods effectiveness is evaluated in terms of the quality of the solutions and the computational effort to obtain



Note: Numbers in parenthesis represent the states (n_{i-1}, t_{i-1})

Optimal Policy for Example 1.
Figure 3.

the solutions. The quality of the solution is only measured by the expected cost of the solution.

In order to test the effectiveness of the approximate methods, two sets of examples were used. The first set is composed of examples 1-3, and compares the approximate methods versus the exact method. Results are shown in Table 4.

To measure the efficiency of the approximate methods we use the following relation

$$\text{Efficiency Method i (\%)} = \frac{[\text{Min}(k_r\ ,\ k_a r_o/t_o)\text{-Expected cost per unit}]_{\text{method i}}}{[\text{Min}(k_r\ ,\ k_a r_o/t_o)\text{-Expected cost per unit}]_{\substack{\text{exact}\\ \text{method}}}} 100 \qquad (5.1)$$

Recall that the Min $(k_r, k_a\ r_o/t_o)$ is the per unit expected cost of the best decision before sampling, the best we can do with our model is given by the denominator of equation (5.1).

Table 1 shows that efficiencies of Methods 1 and 2A are very high so that not much room for improvement remains. Method 2B shows in all three examples a significantly poorer solution.

The policies for Lot 1 generaged by Methods 1 and 2A are fairly close to the optimal policy. Method 2B tends to generage policies with smaller sample sizes.

Execution time is drastically reduced by the approximate methods. Differences among approximate methods are not very significant. However, Method 2B tends to have smaller execution times.

Table 4.

	EXAMPLE	EXACT METHOD	METHOD 1	METHOD 2A	METHOD 2B
$V(r_0,t_0)$	1	1.561	1.573	1.573	1.615
	2	2.593	2.596	2.596	2.634
	3	2.052	2.064	2.063	2.095
Expected cost per unit	1	0.5203	0.5243	0.5243	0.5383
	2	0.8643	0.8653	0.8653	0.878
	3	0.684	0.688	0.6877	0.6983
*Efficiency %	1	-	97.27	97.27	87.7
	2	-	99.26	99.26	89.9
	3	-	86.79	87.88	52.78
Policy for lot 1. (n_1,c_{n_1})	1	(20,1)	(26,1)	(26,1)	(9,0)
	2	(14,0)	(12,0)	(12,0)	(8,0)
	3	(5,0)	(4,0)	(4,0)	(1,0)
Execution time (cp sec.)	1	9.733	0.235	0.24	0.159
	2	8.44	0.19	0.185	0.169
	3	10.171	0.178	0.18	0.16

* Equation (5.1)

Results for Examples 1-3.

Table 5.

Ex.	l	k_I	k_v	k_r	k_a	r_0	t_0	N	E(p)	p_c
4	5	0	0.9	1	11	2	22	50	0.091	0.091
5	5	0	1.1	1	20	1	15	100	0.067	0.05
6	5	0	1.5	1	18	1	20	100	0.05	0.055
7	5	0	1.5	1	10	1	20	100	0.05	0.1

Data for examples 4-7.

Table 6.

Stage i	Number of possible states (NS_i)	
	Example 4	Examples 5-7
1	1	1
2	1326	5151
3	5151	20301
4	11476	45451
5	20301	80601

Number of possible states for examples 4-7.

The second set, Examples 4-7, compares differences among approximate methods. Table 5 shows the data for this set and the number of possible states for each stage i (NS_i). Again the parameters were chosen so that E(p) is above, close and below p_c, and in example 4, $k_v < k_r$ and in Examples 5-7 $k_v > k_r$. From the second part of Table 6 we can appreciate the magnitude of the examples. In Examples 5-7 there is 80,601 possible states at stage 5.

Table 7 presents the results for this set. Again Methods 1 and 2A produce similar expected costs. However, Method 2A shows a slight tendency to obtain smaller expected costs. In all four examples Method 2B produces the worst expected cost. In three out of four examples, Method 2A requires the greatest execution time. And in all examples Method 2B executed in a significantly smaller time.

5.2. Discussion of Results

The main objective of the project was to expand the present Bayesian Theory of sampling inspection by attributes from a static to a dynamic approach, by developing a model with a feedback mechanism that use the information from previous inspection results in the decision making for the current lot.

The model was formulated by assuming a sequence of lots and that the posterior distribtuion of the current lot becomes the prior distribution of the next lot.

We further assumed that the process operates in a random manner with quality p and probability function W(p). However, as we are

Table 7.

	EXAMPLE	METHOD 1	METHOD 2A	METHOD 2B
$V(r_0,t_0)$	4	4.063	4.061	4.089
	5	4.015	4.012	4.064
	6	3.637	3.641	3.721
	7	2.495	2.481	2.499
Expected cost per unit	4	0.8126	0.8122	0.8178
	5	0.8030	0.8024	0.8128
	6	0.7274	0.7282	0.7442
	7	0.4990	0.4962	0.4998
Policy for lot 1 (n_1,c_{n_1})	4	(39,3)	(39,3)	(15,1)
	5	(54,2)	(54,2)	(16,0)
	6	(26,1)	(26,1)	(7,0)
	7	(5,1)	(5,1)	(1,1)
Execution time cp sec.	4	0.935	0.851	0.378
	5	0.613	0.921	0.350
	6	0.426	0.605	0.296
	7	0.328	0.525	0.210

Results for examples 4-7.

collecting the sampling information to update the parameters of W(p) at each stage of the sequence, we also update the estimate of the quality of process, $E_i(p)$, at every stage i. Therefore, a shift of p to say p^0 at some finite stage of the sequence would mean that $E_i(p) = (r_0 + \sum_{j=1}^{i-1} x_j)/(t_0 + \sum_{j=1}^{i-1} n_j)$ tends to p^0 as i tends to infinite. That is, shifts in the process quality are eventually corrected.

After formulating the model most of the research effort was spent in finding a way to solve the model in the most efficient manner. In order to do this we first needed to find an efficient algorithm to solve the single lot case.

A significant contribution was made in this area. If we use the exact method to solve the two following single lot examples.

Table 8

Example	l	k_I	k_v	k_r	k_a	r	t	N	p_c	E(p)
8	1	0	0.28	0.3	3	1	10	1000	0.1	0.1
9	1	0	0.28	0.3	3	2	10	1000	0.1	0.2

Data for examples 8-9

We obtain $(n^*, c_n^*) = (56,5)$, $K'(n^*) = 0.209$ with an execution time of 0.229 cp. sec. for Example 8, and $(n^*, c_n^*) = (75,6)$, $K'(n^*) = 0.284$ with 0.201 cp. sec. for Example 9. These examples as all the computer work of this project was carried out on the University CDC Cyber 170 Model 720.

Moskowitz and Berry (18) report 29.364 sec. and 113.162 sec. on a CDC 6500 for similar examples. These results are not directly comparable as Moskowitz and Berry use a different model in their research, but they certainly give an idea about the efficiency of our algorithm.

After looking at the single lot case we developed the lower bounds of Sections 3.3.2 and 3.4.3, and Theorems 3.5, 3.7 and 3.8 which made it possible to prune our decision tree and made it feasible to solve to optimality sequences of small lots. The effectiveness of these results were discussed in Section 5.1.

The need for approximate methods was clear since the beginning of the project. Three methods were proposed. Method 1 and Method 2A produced better solutions than Method 2B. Differences between Methods 1 and 2A are small, but it appears that Method 2A produces slightly better solutions. In addition, Method 2A fixes the samples sizes of all lots at one level. Therefore, its policies are simpler looking and easier to use in an industrial environment. Method 2B requires the least computation effort, but the loss in the quality of the solutions does not compensate for the computation savings.

The dynamic approach used in this research updates the estimates of the parameters of the prior distribution every time that a sample is taken and inspected. Therefore, our model is a better representation of the inspection process. This, of course, is reflected in a reduction of the per unit expected cost of inspection.

Table 9.

Ex.	Best expected cost per unit for sequence (method)	$K(n^*)$ for single lot (l=1)	Policy for lot 1 in sequence (n_1,c_n)	Optimal policy for single lot (n,c_n)	Expected savings per unit
1	0.5203 (E)	0.567	(20,1)	(9,0)	0.0467
2	0.8643 (E)	0.943	(14,0)	(8,0)	0.0787
3	0.684 (E)	0.712	(5,0)	(1,0)	0.028
4	0.8122 (2A)	0.880	(39,3)	(15,1)	0.0678
5	0.8024 (2A)	0.888	(54,2)	(16,0)	0.0856
6	0.7274 (1)	0.812	(26,1)	(7,0)	0.0846
7	0.4962 (2A)	0.5	(5,1)	(0,0)	0.0038

Note: E = Exact Method
2A= Method 2A
1 = Method 1

Comparison Between a Sequence of Lots and a Single Lot.

Table 9 shows the cost per unit for examples 1-7 for the single lot case (l=1) and compares it versus our previous best result for the given sequence. From Table 9 we can observe that the main factor that influences the expected savings is the amount of sampling required by the optimal policy of the sequence. The more sampling the more savings. This follows as sampling increases our accuracy in estimating p and therefore reduces our risk of wrong decisions. The amount of sampling required is of course a function of the parameters of the model. If, for example, the initial $E(p) = r_o/t_o$ is well above or well below p_c, it is likely that not much sampling is going to be required unless a large sequence of large lots is considered. The same would happen if sampling is costly, that is, if k_I or k_v or both are large compared to k_r or k_a.

5.3 Possible Areas for Future Research

Most of the limitations of this study provide potential areas for further research. In addition, a wide range of extensions to this research are possible.

The cost parameters are assumed to be known constants. In some cases they could be considered as functions instead. For example, the cost of sampling is assumed to be a linear function of n with k_I and k_v as known constants. The variable sampling cost, k_v, may be thought as a function of n perhaps k_v decreasing as n increases in order to differentiate between the per unit variable cost of sampling when n N and the per unit variable cost of 100% inspection.

The term rejection was used collectively for all the possible actions taken on items or lots which are not accepted. Therefore, new models can be developed for different specific situations.

The objective used was the minimization of the total expected cost. For the sampling case, when deriving this quantity a number of indirect effects of sampling inspection were not taken into account. Efforts to include these effects can be of interest.

Inspection is considered 100% effective. This is not true in practice. A more complete model should include the possibility of inspection errors, so that a better representation of the real situation is solved.

The possibility of solving big problems to optimality in a reasonable amount of computer time lies mainly in the development of tight and simple to solve for bounds for the optimal sample size and for the expected cost of sampling. Not much room is left for the improvement of the algorithm for solving the single lot case, but proofs for the results that only have been proven experimentally are desirable.

As our research used single sampling plans, a big field of research is open for models with double, multiple and sequential sampling plans. Restricted Bayesian sampling models could also be used.

BIBLIOGRAPHY

(1) Abel,V., "Some Essentially Complete Classes for Bayesian Multistage Sampling Inspection". Commun. Statist.-Theor. Meth., A9(14), 1483-1489, (1980).

(2) Case,K.E.,Schmidt,J.W. and Bennett,G.K., "Cost Based Acceptance Sampling." Industrial Engineering, 4(11), 26-31, (1972).

(3) Champernowne,D.G., "The Economics of Sequential Sampling Procedures for Defectives". Applied Statistics 2, 118-130, (1953).

(4) Dietrich,D.L. and Sanders,J.L., "A Bayesian Quality Assurance Model for a Multistage Production Process." 1974 ASQC Technical Conference Transactions-Boston, (1974).

(5) Duncan,A.J., "Quality Control and Industrial Statistics.", 4th. Ed., Richard D. Irwin Inc., (1974).

(6) Guenther,W.C., "On the Determination of Single Sampling Attribute Plans Based upon a Linear Cost Model and a Prior Distribution". Technometrics, 13, 3, 483-498, (Aug. 1971).

(7) Hald,A., "The Compound Hypergeometric Distribution and a System of Single Sampling Inspection Plans Based on Prior Distributions and Costs". Technometrics, 2, 275-340, (1960).

(8) Hald,A., "The Determination of Single Sampling Attribute Plans with Given Producer's and Consumer's Risk". Technometrics, 9, 401-415, (1967).

(9) Hald,A., "Asymptotic Properties of Bayesian Single Sampling Plans.", J. Royal Statistics Soc., 29, 162-173, (1967).

(10) Hald,A., "Bayesian Single Sampling Attribute Plans for Continuous Prior Distributions". Technometrics, 10, 667-683, (1968).

(11) Hald,A. and Keiding,N., "Asymtotic Properties of Bayesian Decision Rules for Two Terminal Decisions and Multiple Sampling." I and II. J. Royal Statistics Soc., 31, 455-471 and 34, 55-74, (1969 and 1972).

(12) Hald,A., "Optimum Double Sampling Tests of Given Strength I. The Normal Distribution.", J. Amer. Statist. Ass., 70, 457-462, (1975).

(13) Hald,A., "Statistical Theory of Sampling Inspection by Attributes". Academic Press. (1981).

(14) Lindley,D.V. and Barnett,V.D., "Sequential Sampling: Two Decision Problems with Linear Losses for Binomial and Normal Random Variables.", Biometrika, 52, 507-537, (1965).

(15) Martz,H.F., "Empirical Bayes Single Sampling Plans for Specified Posterior Consumer and Producer Risks.", Naval Research Logistics Quarterly, vol.22, 4, 651-665, (Dec. 1975).

(16) Mood, A.M., "On the Dependence of Sampling Inspection Plans upon Population Distributions. ", Annals of Mathematical Statistics, 14, 415-425, (1943).

(17) Moreno,C., "A Performance Approach to Attribute Sampling and Multiple Action Decisions. ", AIIE Transactions, vol. 11, 3, 183-197, (Sep. 1979).

(18) Moskowitz, H. and Berry,W.L., "A Bayesian Algorithm for Determining Optimal Single Sample Acceptance Plans for Product Attributes.", Managment Science, 22, 11, 1238-1250, (Sep. 1976).

(19) Pfanzagl,J., "Sampling Procedures Based on Prior Distributions and Costs", Technometrics, 5, 47-61, (1963).

(20) Raiffa,H. and Schlaifer,R., "Applied Statistical Decision Theory.", Division of Research, Graduate School of Business Administration, Harvard University, Boston, (1961).

(21) Sculer,W., "Multistage Sampling Procedures Based on Prior Distributions and Costs.", Ann. Math. Stat. ,38, 464-470, (1967).

(22) Smith,B.E., "The Economics of Sampling Inspection.", Industrial Quality Control, 453-458, (Mar. 1965).

(23) Stacy,E.W., Hunsinger,L.E. and Price,J.F., "Determining Economic Sampling Plans.", IBM Systems J., 8, 220-244, (1969).

(24) Stewart,R.D., Montgomery,D.C. and Heikes R.G., "Choice of Double Sampling Plans Based on Prior Distributions and Costs.", AIIE Transactions, vol. 10, 1, 19-30, (Mar. 1978).

(25) Wagner,H.M., "Principles of Operations Research: with Applications to Managerial Decisions.", 2nd. Ed., Prentice Hall, (1975).

(26) Wetherill,G.B. and Chiu,W.K., "A Review of Acceptance Sampling Schemes with Emphasis on the Economic Aspect.", Int. Statis. Rev., vol.43, 2, 191-210, (1975).

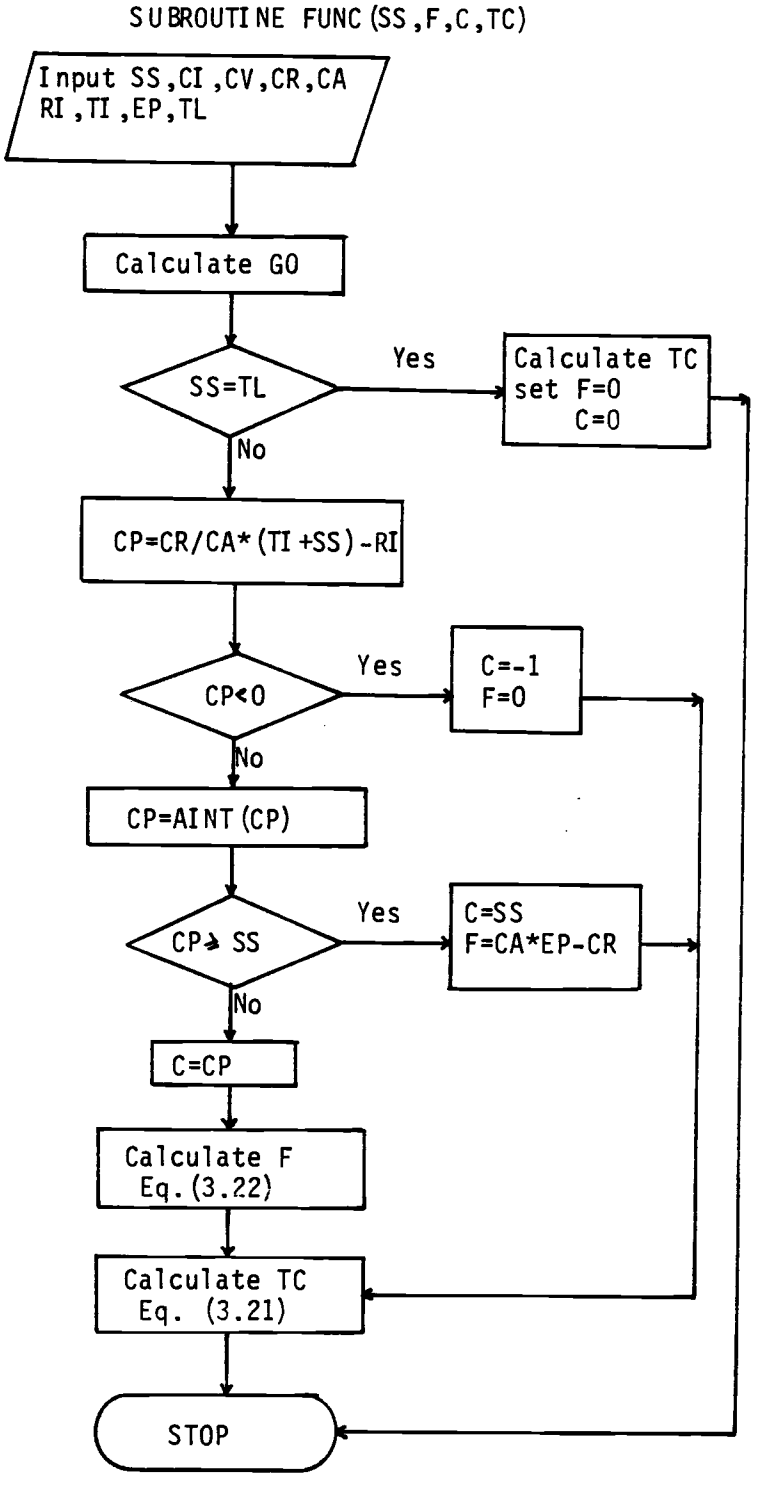
(27) Wortham,A.W. and Wilson,E.B., "A Backward Recursive Technique for Optimal Sequential Sampling Plans.", Naval Research Logistics Quarterly, 18, 12, 203-213, (1971).

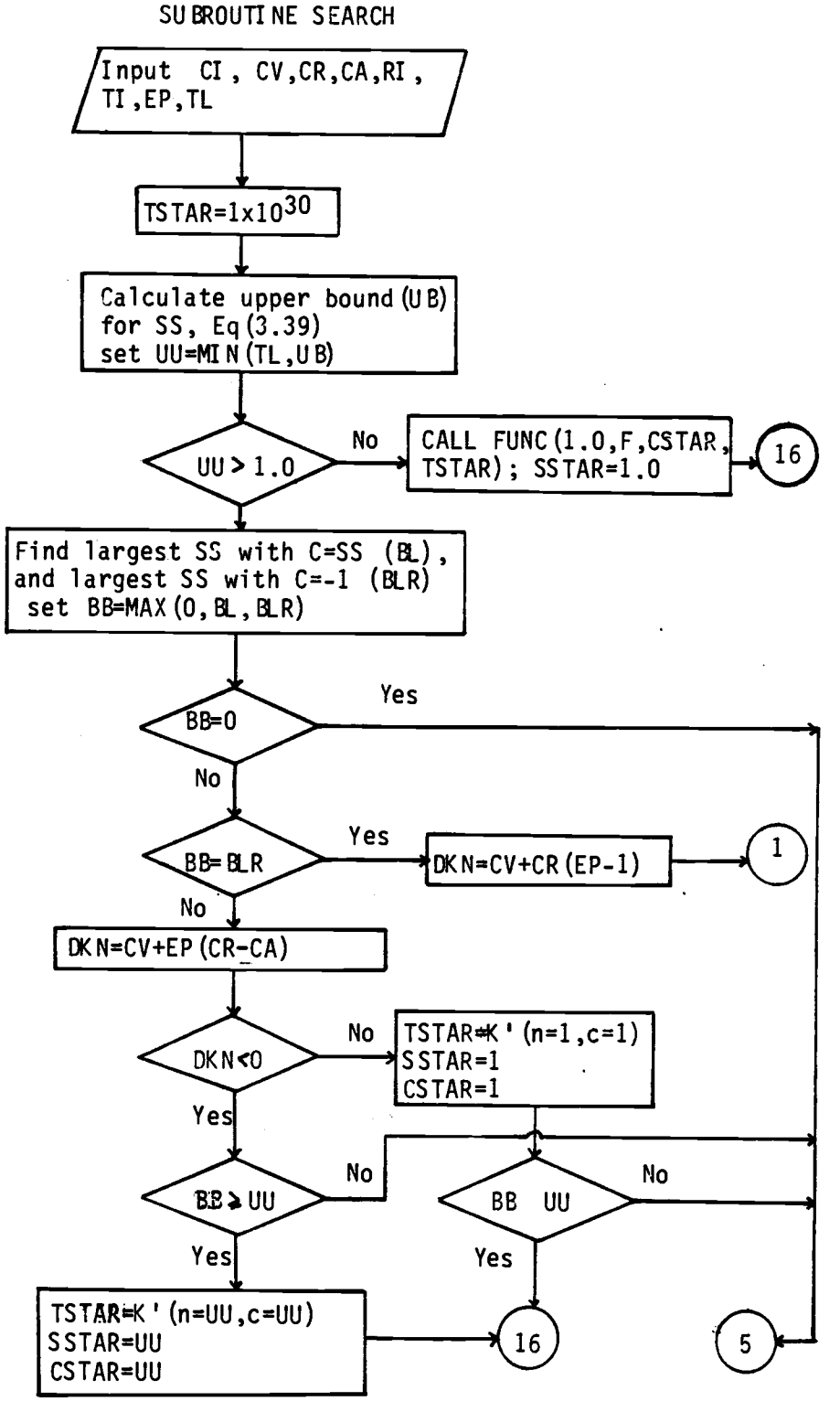
APPENDIX

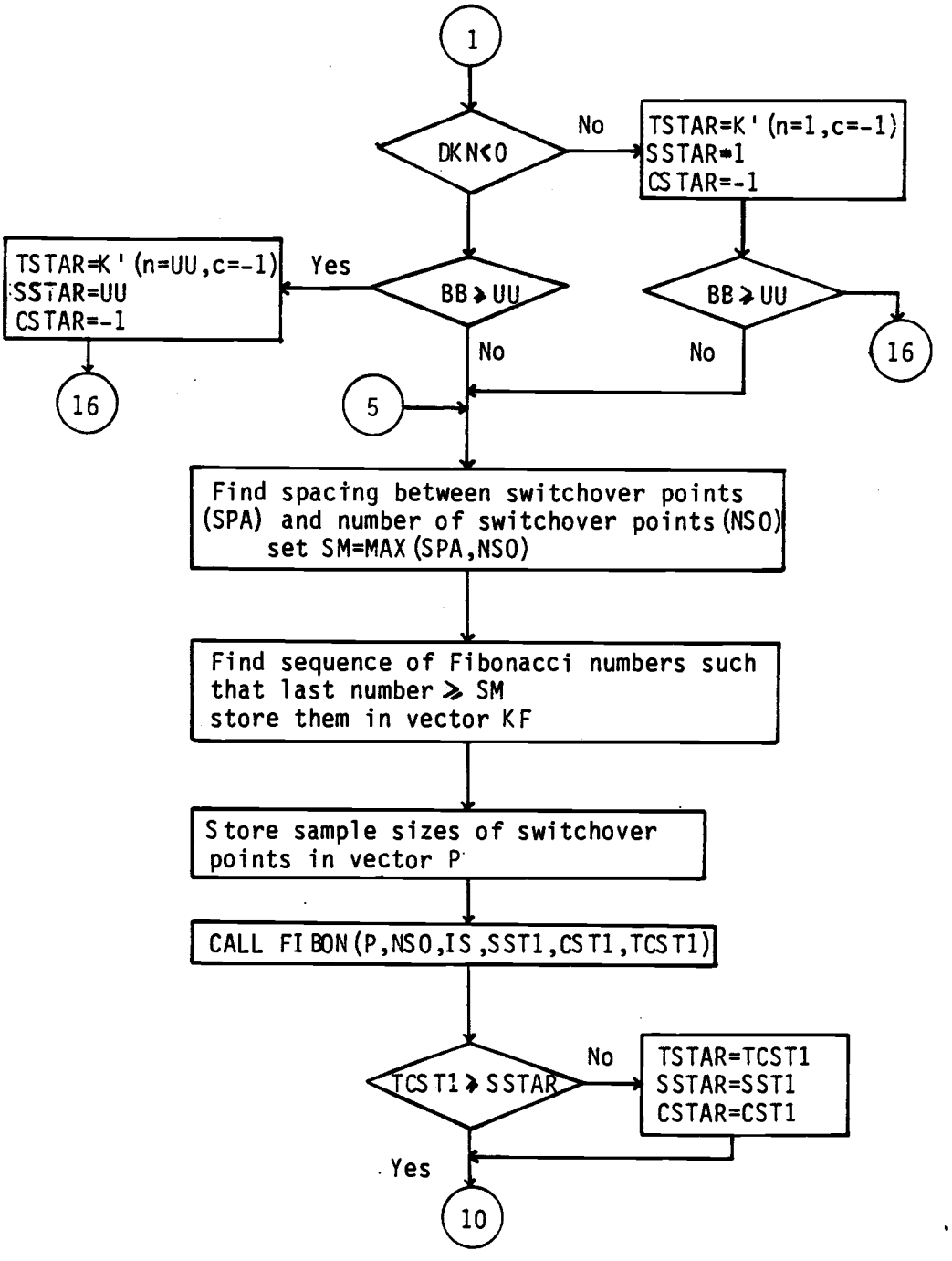
This Appendix is divided in two main parts. Firstly, flow charts of the four methods and subroutines are given. And the second part contains the programs listings.

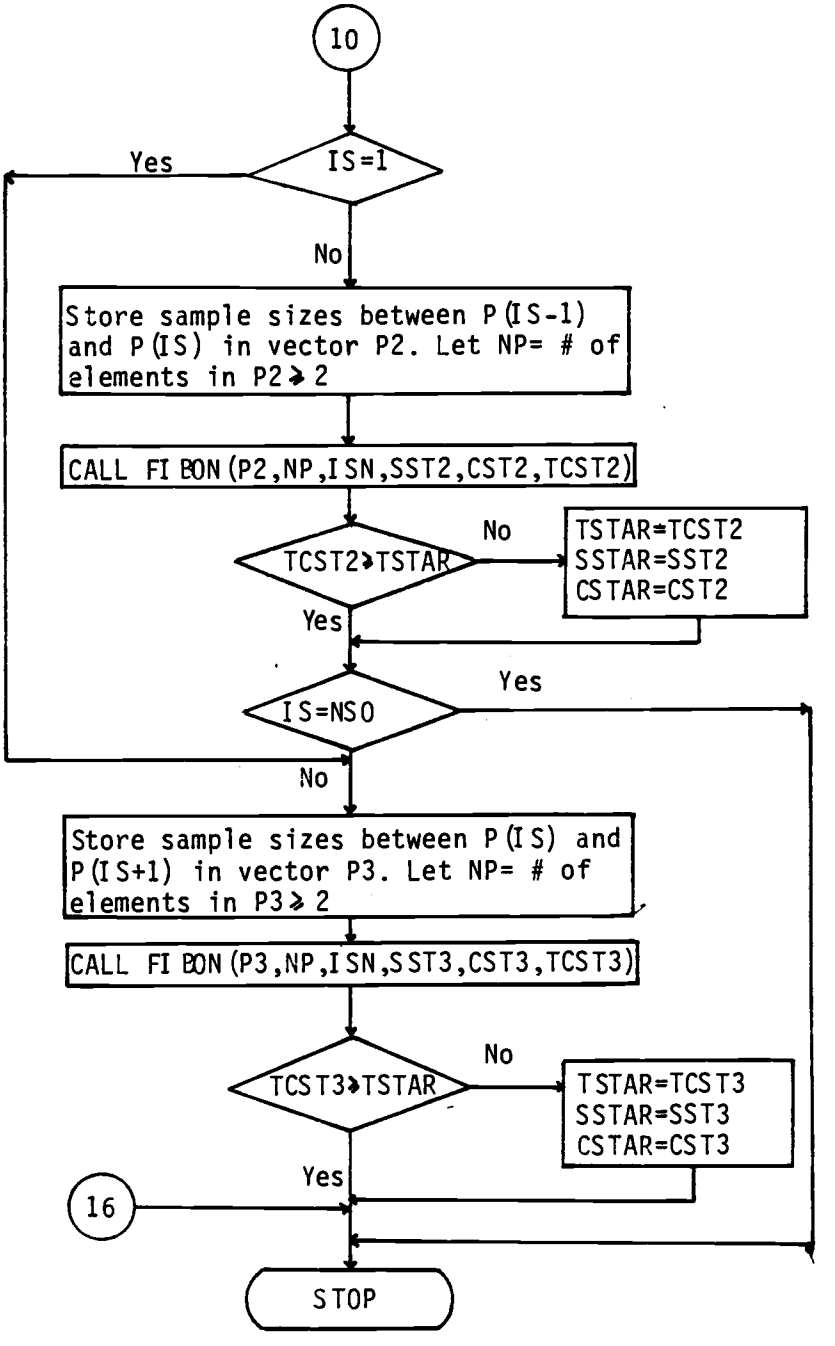
All computer work was done in Fortran IV. Throughout the Appendix the following notation will be used:

CI=k_I	CV=k_v
CR=k_r	CA=k_a
RI=r_{i-1}	TI=t_{i-1}
EP=E(p)	TL=N
SS=n_i	SSTAR=n_i^*
F=F(n_i)	C=c_n
CSTAR=c_n^*	
TC=K(n_i)	
DKN=$_nK(n_i)$	
GO=$g_n(0)$	
TES=$k_v + k_r(t_{i-1}-r_{i-1})/t_{i-1}$	
NL=1	
OHI=K(N)	
AWI=$k_a E(p)$	
BS(I,J)=LBS(r_{i-1},t_{i-1})	
BSS=LBS_i	

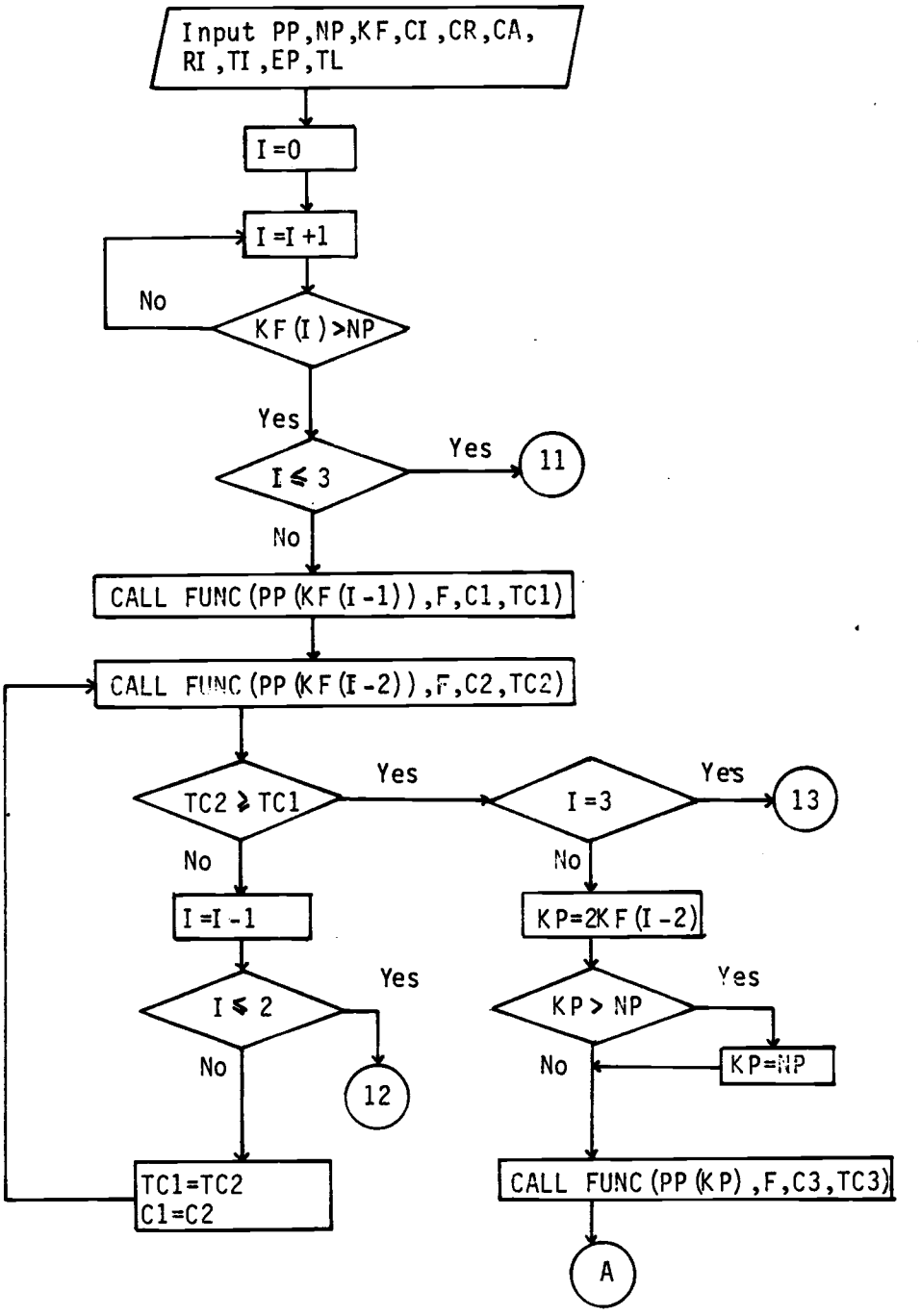

SUBROUTINE FUNC(SS,F,C,TC)
Input SS,CI,CV,CR,CA
RI,TI,EP,TL
Calculate GO
SS=TL
Yes
Calculate TC
set F=0
C=0
No
CP=CR/CA*(TI+SS)-RI
CP<0
Yes
C=-1
F=0
No
CP=AINT(CP)
CP≥ SS
Yes
C=SS
F=CA*EP-CR
No
C=CP
Calculate F
Eq.(3.22)
Calculate TC
Eq. (3.21)
STOP

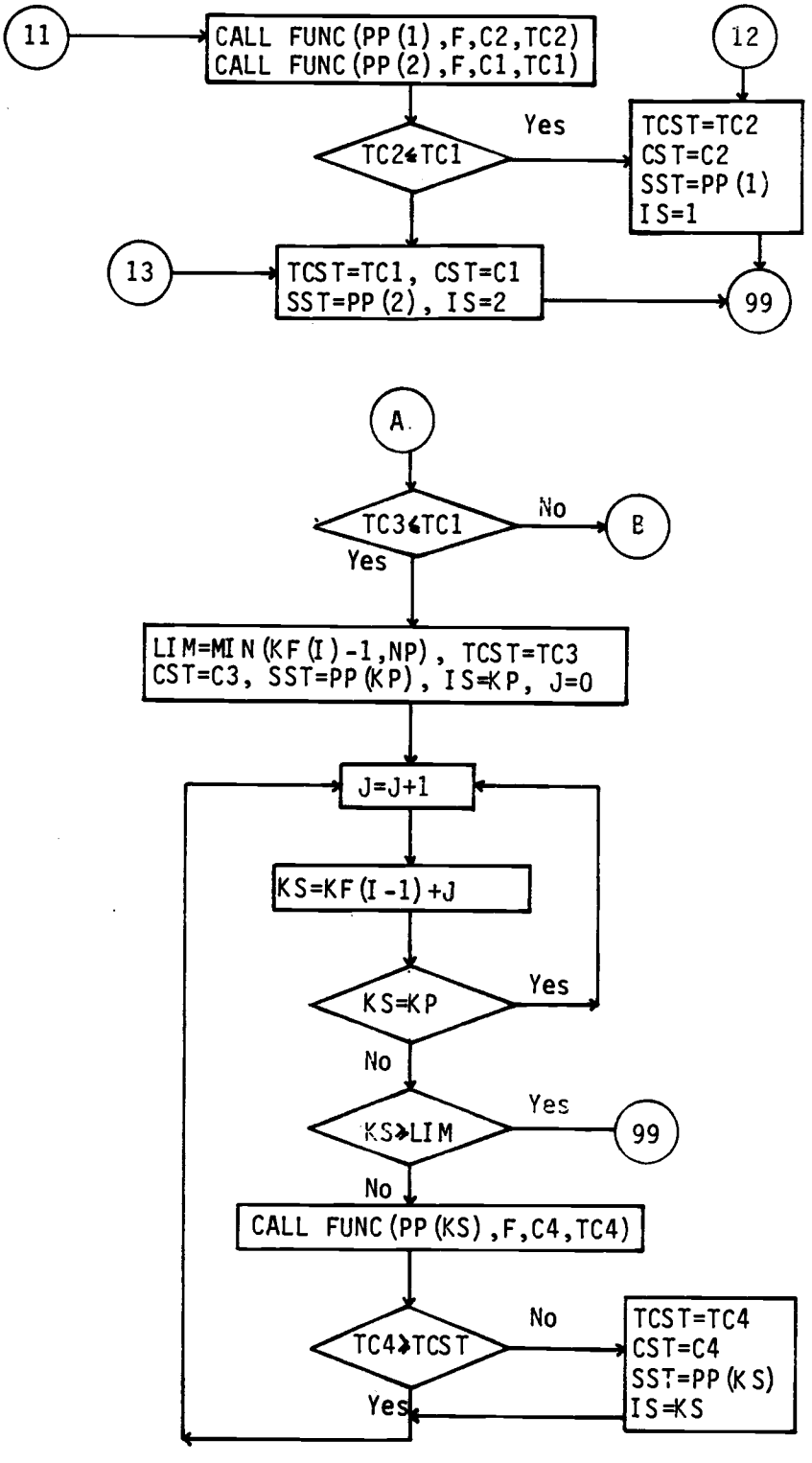

SUBROUTINE SEARCH
Input CI, CV,CR,CA,RI, TI,EP,TL
TSTAR=1x10^30
Calculate upper bound (UB) for SS, Eq (3.39) set UU=MIN (TL,UB)
UU > 1.0
No
CALL FUNC (1.0,F,CSTAR, TSTAR); SSTAR=1.0
16
Find largest SS with C=SS (BL), and largest SS with C=-1 (BLR) set BB=MAX (0, BL, BLR)
BB=0
Yes
No
BB= BLR
Yes
DKN=CV+CR (EP-1)
1
No
DKN=CV+EP (CR-CA)
DKN<0
No
TSTAR=K' (n=1,c=1) SSTAR=1 CSTAR=1
Yes
BB ≥ UU
No
BB UU
No
Yes
Yes
TSTAR=K' (n=UU,c=UU) SSTAR=UU CSTAR=UU
16
5

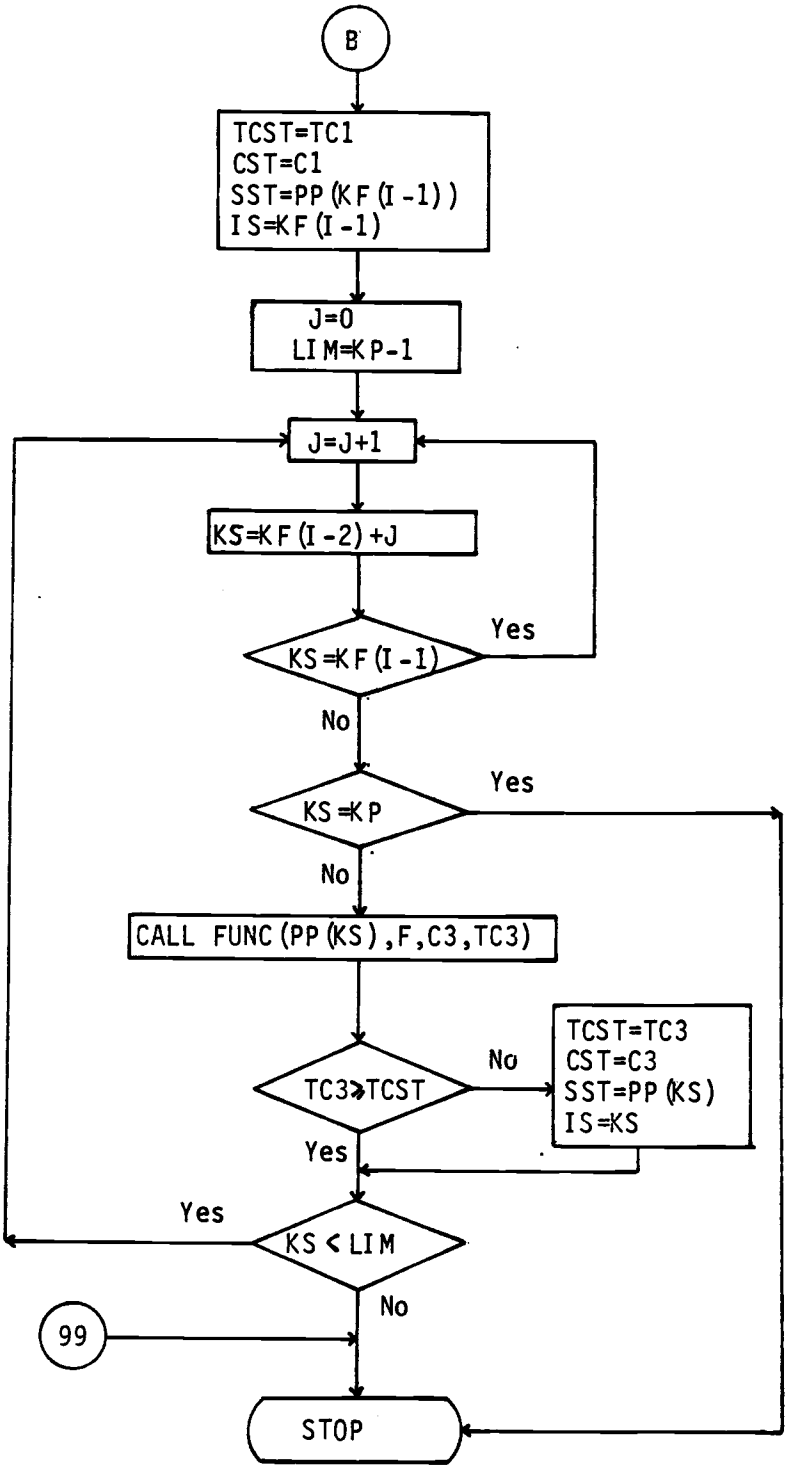

1
DKN<0
No
TSTAR=K'(n=1,c=-1)
SSTAR=1
CSTAR=-1
TSTAR=K'(n=UU,c=-1)
SSTAR=UU
CSTAR=-1
Yes
BB≥UU
BB≥UU
No
No
16
16
5
Find spacing between switchover points (SPA) and number of switchover points (NSO)
set SM=MAX(SPA,NSO)
Find sequence of Fibonacci numbers such that last number ≥ SM
store them in vector KF
Store sample sizes of switchover points in vector P
CALL FIBON(P,NSO,IS,SST1,CST1,TCST1)
TCST1≥SSTAR
No
TSTAR=TCST1
SSTAR=SST1
CSTAR=CST1
Yes
10


10
IS=1
Yes
No
Store sample sizes between P(IS-1) and P(IS) in vector P2. Let NP= # of elements in P2 ≥ 2
CALL FIBON(P2,NP,ISN,SST2,CST2,TCST2)
TCST2≥TSTAR
No
TSTAR=TCST2
SSTAR=SST2
CSTAR=CST2
Yes
IS=NSO
Yes
No
Store sample sizes between P(IS) and P(IS+1) in vector P3. Let NP= # of elements in P3 ≥ 2
CALL FIBON(P3,NP,ISN,SST3,CST3,TCST3)
TCST3≥TSTAR
No
TSTAR=TCST3
SSTAR=SST3
CSTAR=CST3
Yes
16
STOP

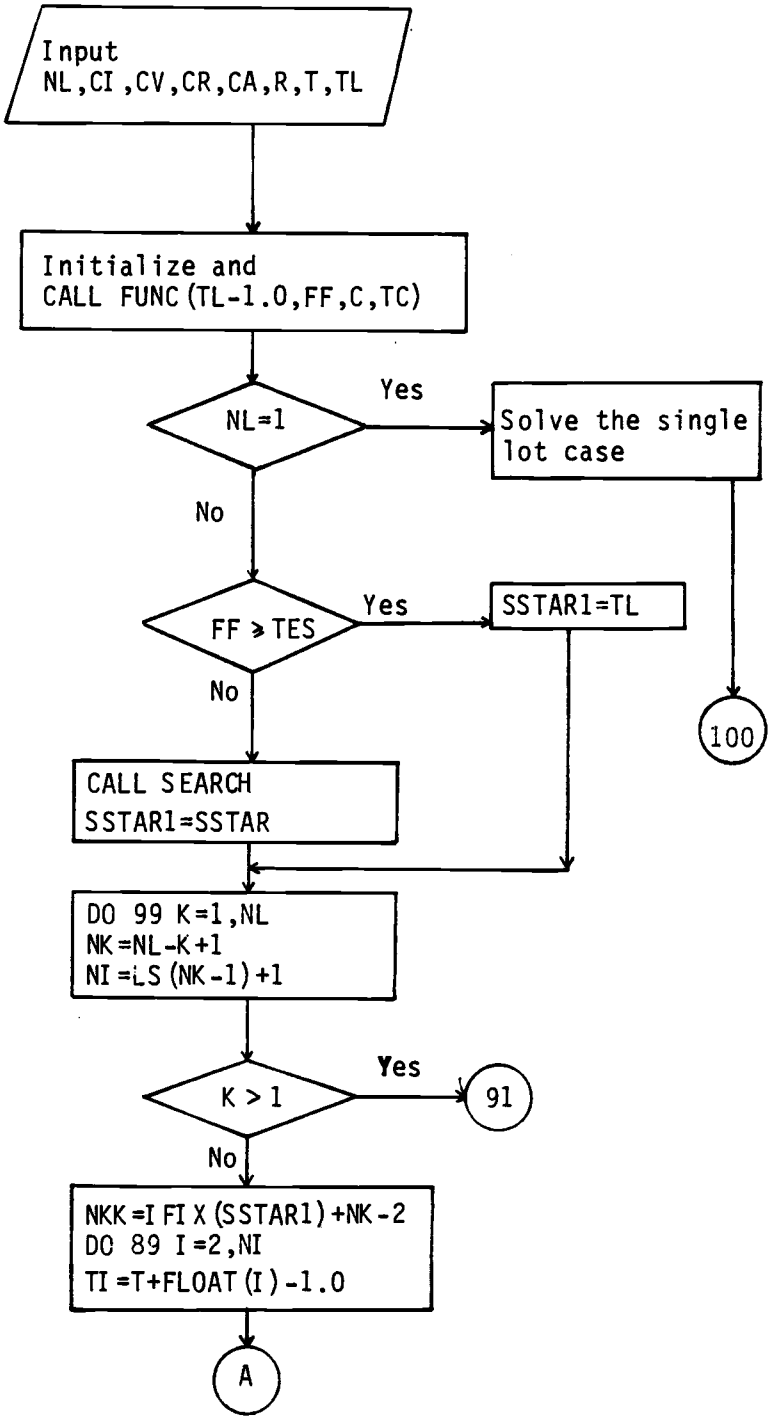
SUBROUTINE FIBON(PP,NP,IS,SST,CST,TCST)

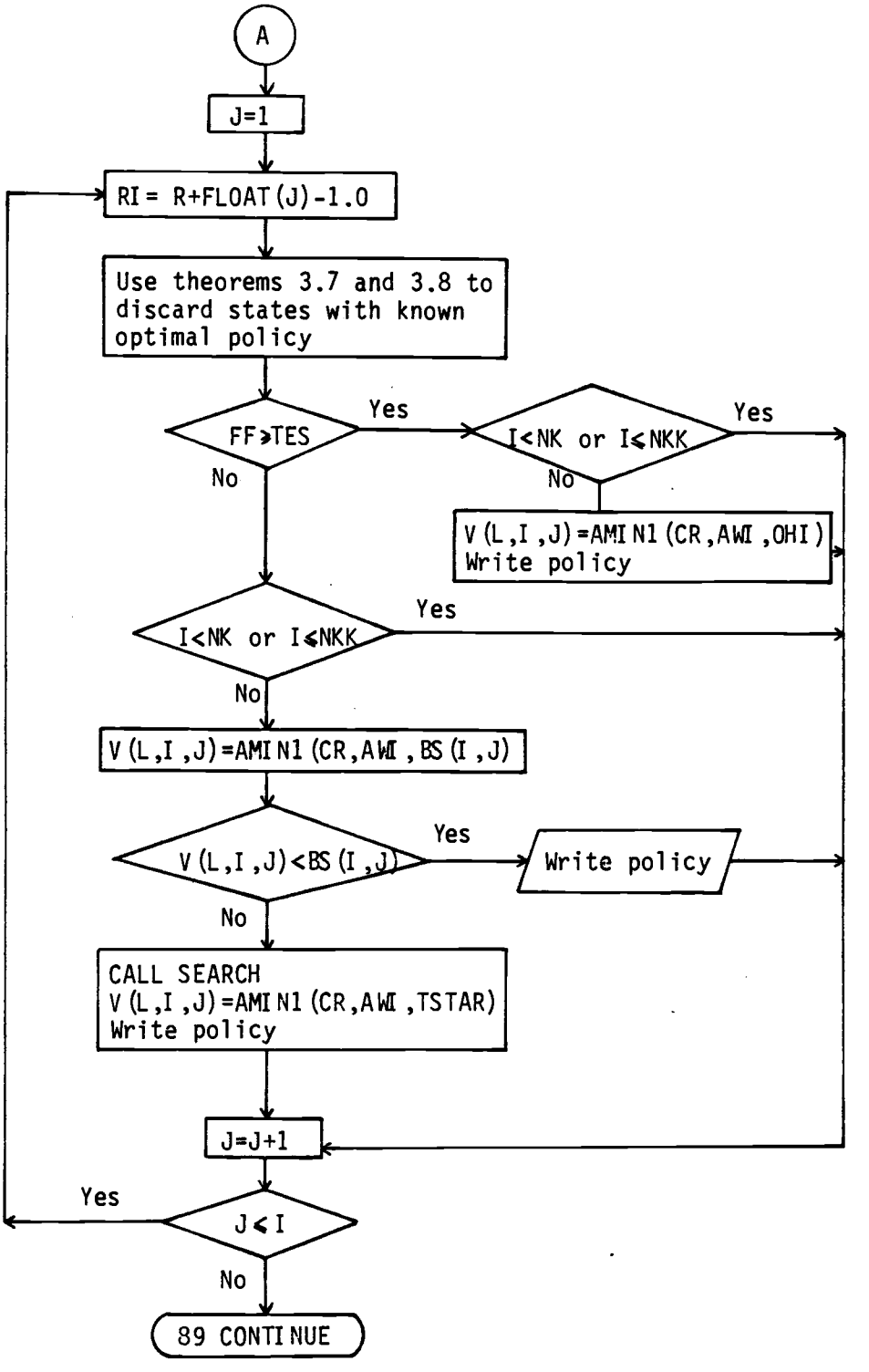


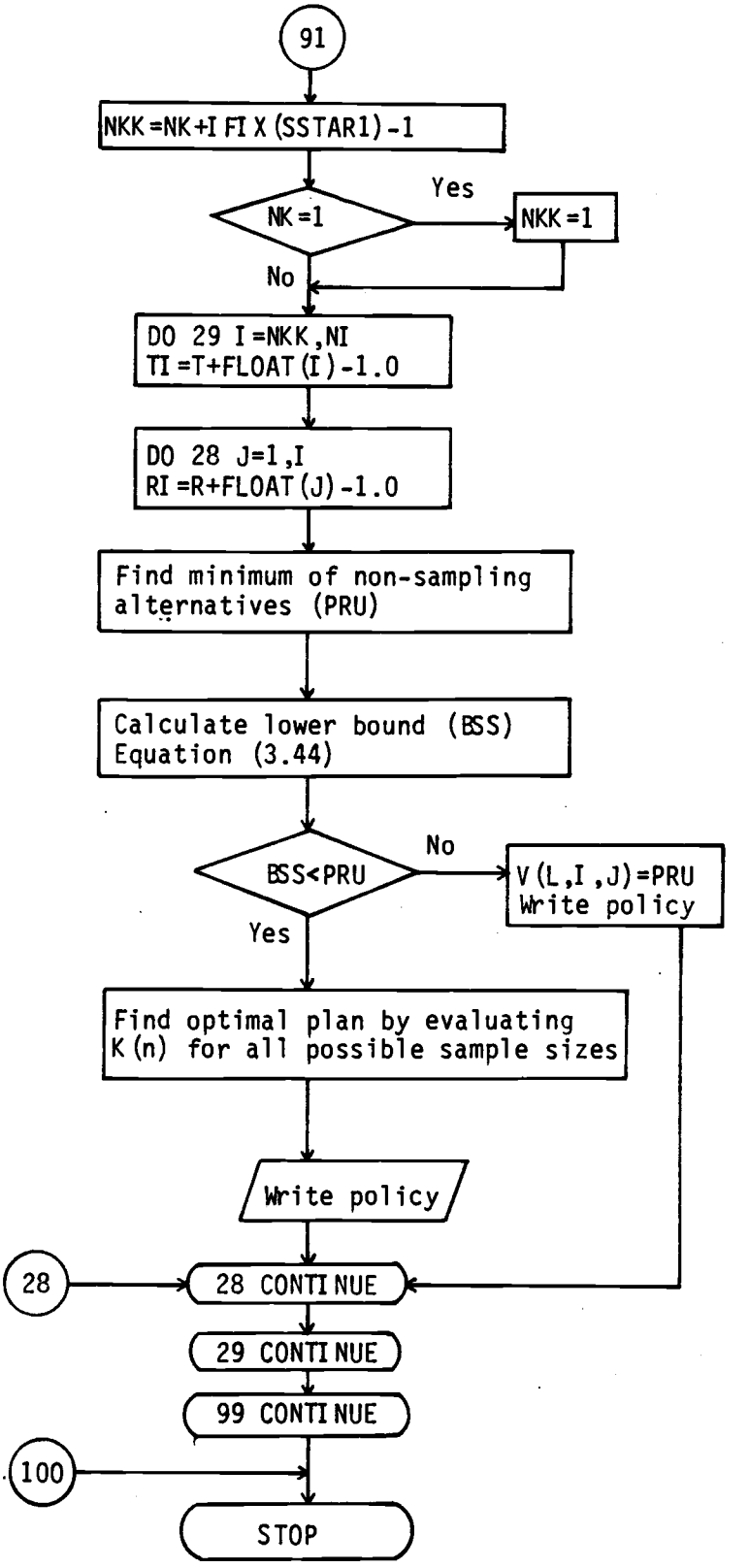

11
CALL FUNC(PP(1),F,C2,TC2)
CALL FUNC(PP(2),F,C1,TC1)
12
TC2<TC1
Yes
TCST=TC2
CST=C2
SST=PP(1)
IS=1
13
TCST=TC1, CST=C1
SST=PP(2), IS=2
99
A
TC3<TC1
No
B
Yes
LIM=MIN(KF(I)-1,NP), TCST=TC3
CST=C3, SST=PP(KP), IS=KP, J=0
J=J+1
KS=KF(I-1)+J
KS=KP
Yes
No
KS>LIM
Yes
99
No
CALL FUNC(PP(KS),F,C4,TC4)
TC4>TCST
No
TCST=TC4
CST=C4
SST=PP(KS)
IS=KS
Yes


B
TCST=TC1
CST=C1
SST=PP(KF(I-1))
IS=KF(I-1)
J=0
LIM=KP-1
J=J+1
KS=KF(I-2)+J
KS=KF(I-1)
Yes
No
KS=KP
Yes
No
CALL FUNC(PP(KS),F,C3,TC3)
TC3>TCST
No
TCST=TC3
CST=C3
SST=PP(KS)
IS=KS
Yes
KS < LIM
Yes
No
99
STOP

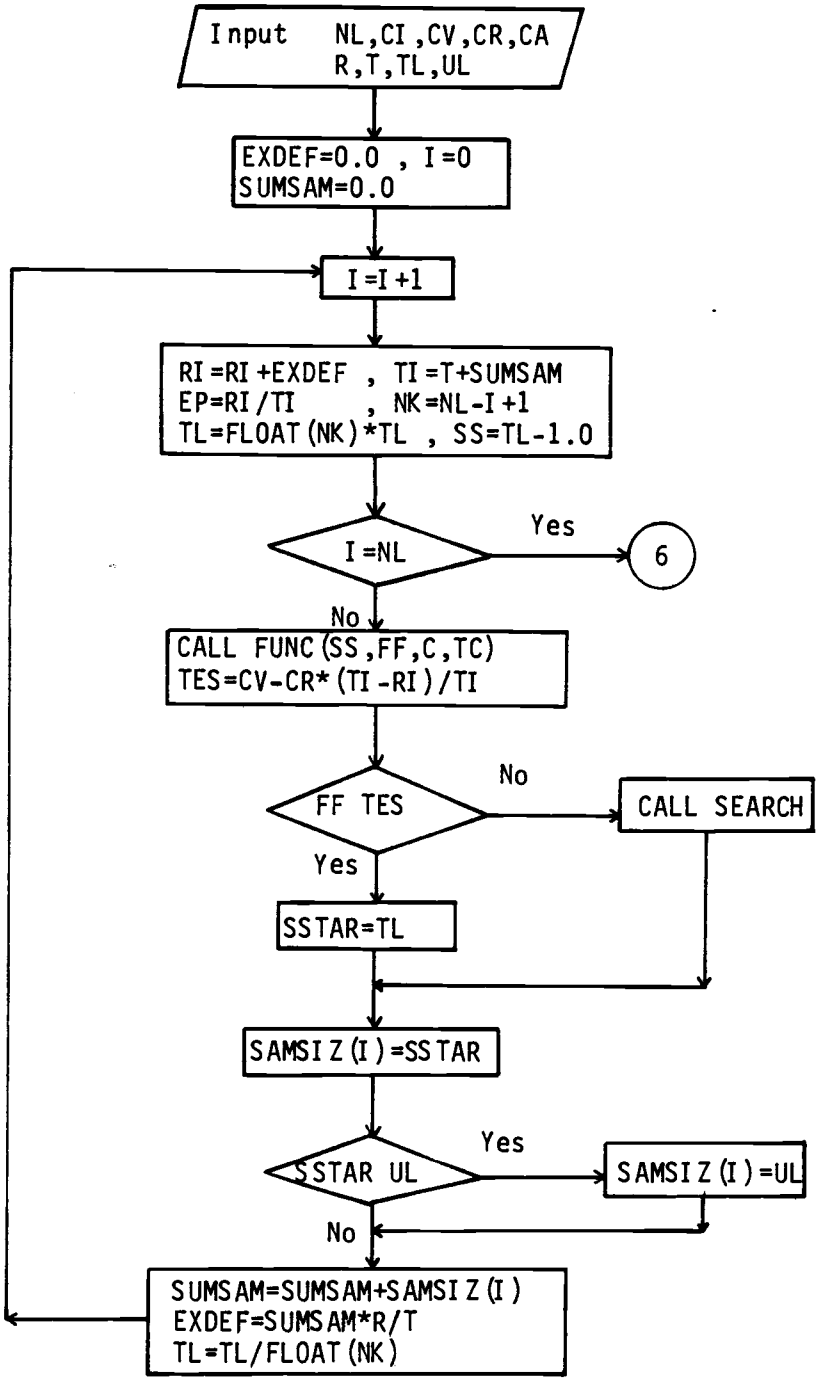
EXACT METHOD

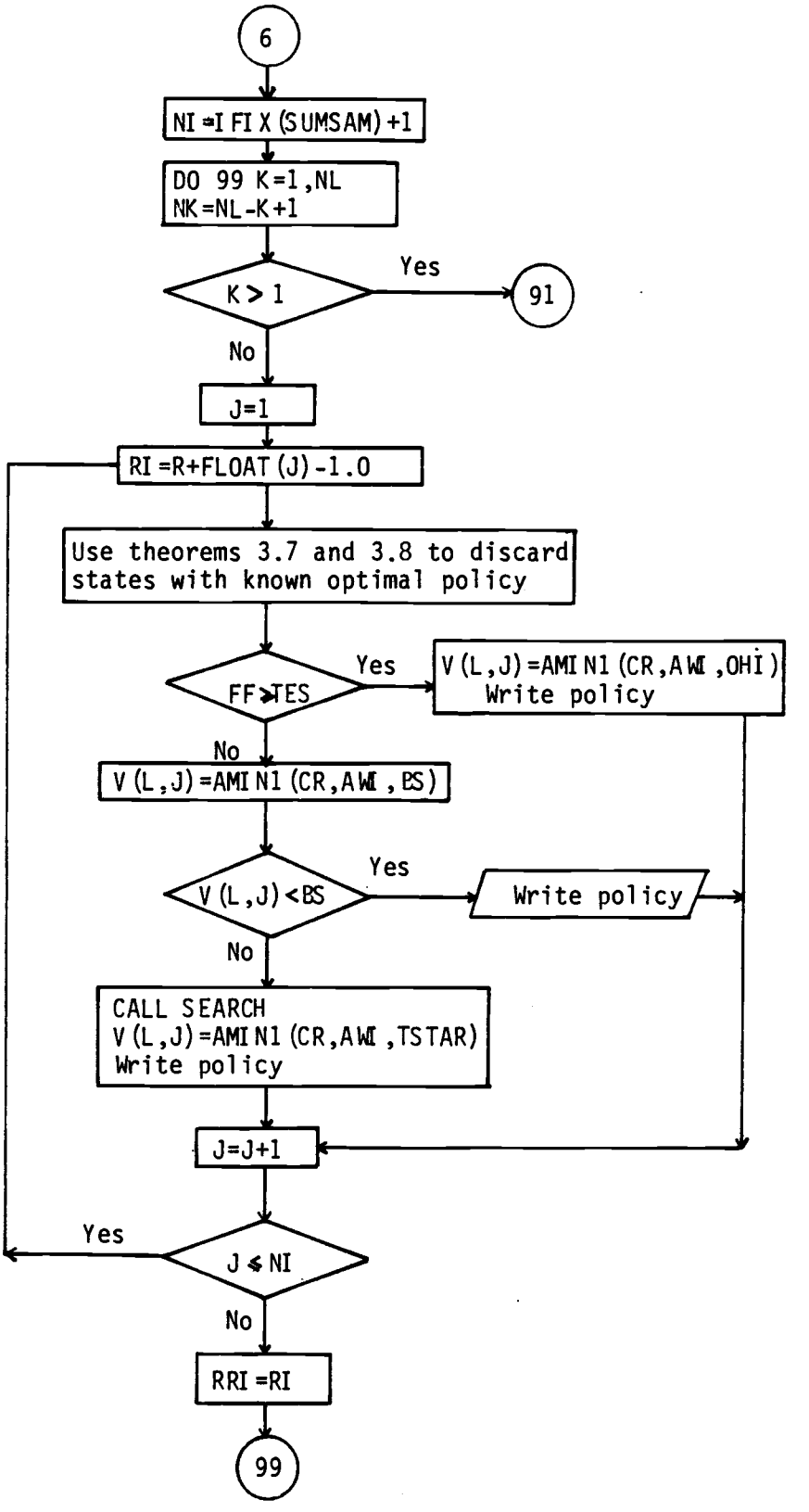


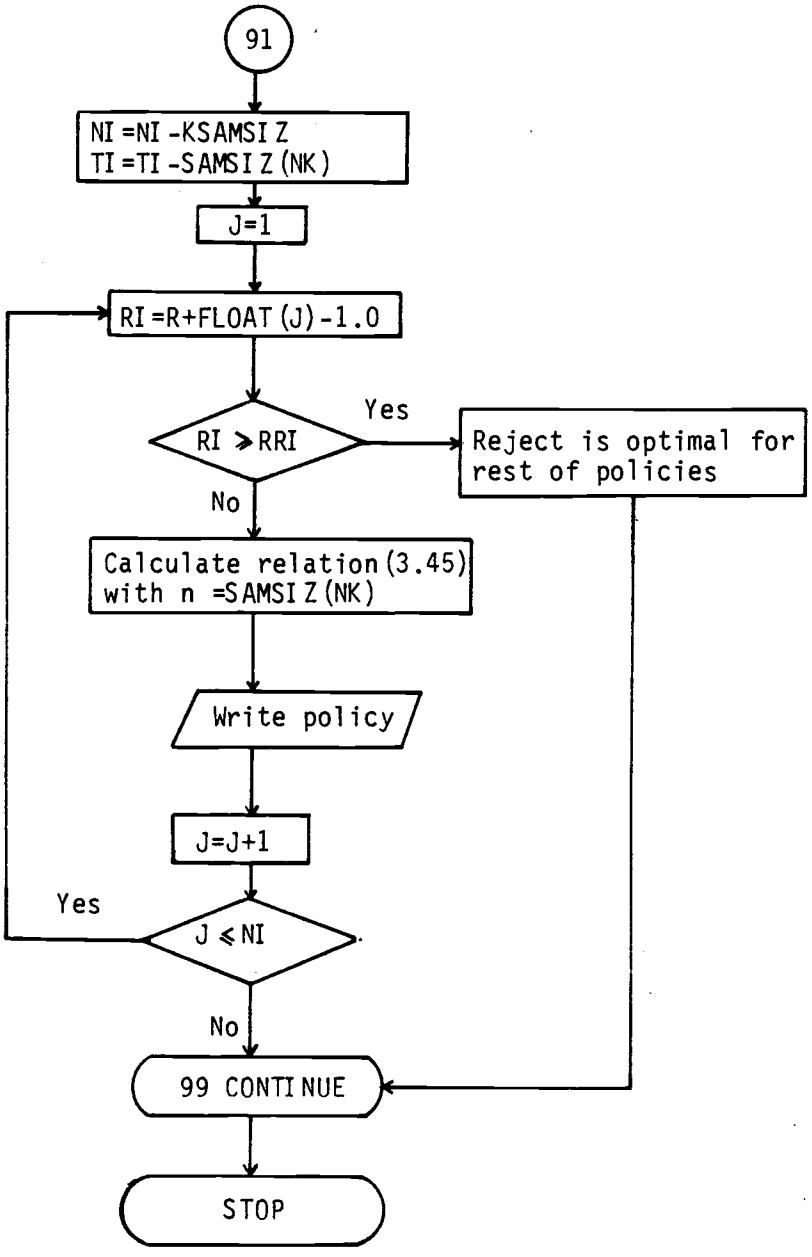

A
J=1
RI = R+FLOAT(J)-1.0
Use theorems 3.7 and 3.8 to discard states with known optimal policy
FF≥TES
Yes
No
I<NK or I≤NKK
Yes
No
V(L,I,J)=AMIN1(CR,AWI,OHI)
Write policy
I<NK or I≤NKK
Yes
No
V(L,I,J)=AMIN1(CR,AWI,BS(I,J)
V(L,I,J)<BS(I,J)
Yes
Write policy
No
CALL SEARCH
V(L,I,J)=AMIN1(CR,AWI,TSTAR)
Write policy
J=J+1
J<I
Yes
No
89 CONTINUE


91
NKK=NK+IFIX(SSTAR1)-1
NK=1
Yes
NKK=1
No
DO 29 I=NKK,NI
TI=T+FLOAT(I)-1.0
DO 28 J=1,I
RI=R+FLOAT(J)-1.0
Find minimum of non-sampling alternatives (PRU)
Calculate lower bound (BSS) Equation (3.44)
BSS<PRU
No
V(L,I,J)=PRU
Write policy
Yes
Find optimal plan by evaluating K(n) for all possible sample sizes
Write policy
28
28 CONTINUE
29 CONTINUE
99 CONTINUE
100
STOP

METHOD 1


Input NL,CI,CV,CR,CA R,T,TL,UL
EXDEF=0.0 , I=0
SUMSAM=0.0
I=I+1
RI=RI+EXDEF , TI=T+SUMSAM
EP=RI/TI , NK=NL-I+1
TL=FLOAT(NK)*TL , SS=TL-1.0
I=NL
Yes
6
No
CALL FUNC(SS,FF,C,TC)
TES=CV-CR*(TI-RI)/TI
FF TES
No
CALL SEARCH
Yes
SSTAR=TL
SAMSIZ(I)=SSTAR
SSTAR UL
Yes
SAMSIZ(I)=UL
No
SUMSAM=SUMSAM+SAMSIZ(I)
EXDEF=SUMSAM*R/T
TL=TL/FLOAT(NK)


6
NI=IFIX(SUMSAM)+1
DO 99 K=1,NL
NK=NL-K+1
K > 1
Yes
91
No
J=1
RI=R+FLOAT(J)-1.0
Use theorems 3.7 and 3.8 to discard states with known optimal policy
FF≥TES
Yes
V(L,J)=AMIN1(CR,AWI,OHI)
Write policy
No
V(L,J)=AMIN1(CR,AWI,BS)
V(L,J)<BS
Yes
Write policy
No
CALL SEARCH
V(L,J)=AMIN1(CR,AWI,TSTAR)
Write policy
J=J+1
J ≤ NI
Yes
No
RRI=RI
99


91
NI =NI -KSAMSI Z
TI =TI -SAMSI Z (NK)
J=1
RI =R+FLOAT (J) -1.0
RI ⩾ RRI
Yes
Reject is optimal for rest of policies
No
Calculate relation (3.45) with n =SAMSI Z (NK)
Write policy
J=J+1
Yes
J ⩽ NI
No
99 CONTI NUE
STOP

```
      PROGRAM BAYES2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     EXACT METHOD.
C
      COMMON /DATA1/ CI,CV,CR,CA,RI,TI,EP,TL,G0
      COMMON /DATA4/ TSTAR,CSTAR,SSTAR
      DIMENSION V(2,101,101),BS(101,101)
      READ (5,1) NL,CI,CV,CR,CA,R,T,TL
    1 FORMAT (I5,7F7.2)
      WRITE (6,2)
    2 FORMAT (2X,/)
      WRITE(6,4)
    4 FORMAT(4X,"L    KI      KV      KR      KA      R0      T0      N")
      WRITE (6,1) NL,CI,CV,CR,CA,R,T,TL
      WRITE (6,2)
      CS=CI/TL+CV
      RI=R
      TI=T
      EP=RI/TI
      OHI=CS+CR*EP
      CALL FUNC(TL-1.0,FF,C,TC)
      TES=CV-CR*(TI-RI)/TI
      IF(NL.EQ.1) GO TO 71
      IF(FF.GE.TES) GO TO 48
      BS(1,1)=OHI+(1.0-1.0/TL)*(FF-TES)
      CALL SEARCH
      SSTAR1=SSTAR
      GO TO 49
   48 BS(1,1)=OHI
      SSTAR1=TL
   49 LS=IFIX(TL)
      CRA=CR/CA
      ULP2=1-CV/CR
      SLP2=CV/(CA-CR)
      L=1
      DO 99 K=1,NL
      NK=NL-K+1
      WRITE (6,2)
      WRITE (6,3) NK
    3 FORMAT (2X,"STAGE NUMBER:",I4)
      WRITE (6,2)
      NI=LS*(NK-1)+1
      IF(L.EQ.2) GO TO 5
      L=2
      M=1
      GO TO 6
    5 L=1
      M=2
    6 IF(K.GT.1) GO TO 91
      WRITE (6,14)
   14 FORMAT(2X,"    RI        TI       REJECT  ACCEPT ",
     1"SAMCO/LB  SAMSIZ  ACCNUM  V( )")
      WRITE(6,2)
      SS=TL-1.0
      NKK=IFIX(SSTAR1)+NK-2
      DO 89 I=2,NI
      TI=T+FLOAT(I)-1.0
      ULP1=CRA*(1+TL/TI)
      SLP1=TL/TI*(CRA-1)+CRA
      AMAXEP=AMAX1(ULP1,ULP2)
      AMINEP=AMIN1(SLP1,SLP2)
      J=1
   27 RI=R+FLOAT(J)-1.0
      EP=RI/TI
      IF(EP.LT.AMINEP) GO TO 21
      IF(EP.GT.AMAXEP) GO TO 24
      AMI=CA*EP
      OHI=CS+CR*EP
      CALL FUNC(SS,FF,C,TC)
      TES=CV-CR*(TI-RI)/TI
      IF(FF.GE.TES) GO TO 8
      BS(I,J)=OHI+(1.0-1.0/TL)*(FF-TES)
      IF(I.LT.NK.OR.I.LE.NKK) GO TO 88
      V(L,I,J)=AMIN1(CR,AMI,BS(I,J))
      IF(V(L,I,J).LT.BS(I,J)) GO TO 10
      CALL SEARCH
      V(L,I,J)=AMIN1(CR,AMI,TSTAR)
```

```
      WRITE (6,17) RI,TI,CR,AWI,TSTAR,SSTAR,CSTAR,V(L,I,J)
   17 FORMAT (2X,"OPTIMAL SAMPLE CALCULATED",8F8.3)
      GO TO 88
   10 WRITE (6,19) RI,TI,CR,AWI,BS(I,J),V(L,I,J)
   19 FORMAT (27X,5F8.3,16X,F8.3)
      GO TO 88
    8 BS(I,J)=-2.0
      IF(I.LT.NK.OR.I.LE.NKK) GO TO 88
      V(L,I,J)=AMIN1(CR,AWI,OHI)
      WRITE(6,20) RI,TI,CR,AWI,OHI,V(L,I,J)
   20 FORMAT(2X,"100% SAMPLING",12X,5F8.3,16X,F8.3)
      GO TO 88
   21 JA=IFIX(TI*AMINEP-R+1.0)
      J=MINO(JA,I)
      DO 22 KL=1,J
      RII=R+FLOAT(KL)-1.0
      V(L,I,KL)=CA*RII/TI
   22 BS(I,KL)=0.0
      WRITE(6,23) RI,RII,TI
   23 FORMAT(2X,"ACCEPT WITHOUT SAMPLING IS OPTIMAL FROM RI=",
     3F8.3," TO RI=",F8.3," FOR TI=",F8.3)
      GO TO 88
   24 DO 25 KL=J,I
      V(L,I,KL)=CR
   25 BS(I,KL)=-1.0
      J=I
      RII=R+FLOAT(I)-1.0
      WRITE(6,26) RI,RII,TI
   26 FORMAT(2X,"REJECT WITHOUT SAMPLING IS OPTIMAL FROM RI=",
     4F8.3," TO RI=",F8.3," FOR TI=",F8.3)
   88 J=J+1
      IF(J.LE.I) GO TO 27
   89 CONTINUE
      GO TO 99
   91 WRITE (6,14)
      WRITE(6,2)
      NKK=NK+IFIX(SSTAR1)-1
      IF(NK.EQ.1) NKK=1
      DO 29 I=NKK,NI
      TI=T+FLOAT(I)-1.0
      DO 28 J=1,I
      RI=R+FLOAT(J)-1.0
      EP=RI/TI
      AWII=FLOAT(K)*CA*EP
      RWII=FLOAT(K)*CR
      PRU=AMIN1(AWII,RWII)
      IF(BS(I,J).EQ.0.0) BS(I,J)=(CI+CV+CA*EP*(TL-1.0)+CR*EP)/TL
      IF(BS(I,J).EQ.-1.0) BS(I,J)=(CI+CV+CR*(TL-1.0+EP))/TL
      CALL FUNC(TL,F,C,TC)
      SUM=V(M,I+LS,J)*G0
      X=1.0
      G=G0
   37 KX=IFIX(X)
      G=(RI+X-1.0)*(TL-X+1.0)/(X*(TL+TI-RI-X))*G
      SUM=SUM+V(M,I+LS,J+KX)*G
      X=X+1.0
      IF(X.LE.TL) GO TO 37
      TCSTAR=TC+SUM
      STAR=TL
      CSTAR=C
      IF(BS(I,J).EQ.-2.0) GO TO 34
      BSS=BS(I,J)+SUM
      IF(BSS.LT.PRU) GO TO 30
      V(L,I,J)=PRU
      WRITE (6,19) RI,TI,RWII,AWII,BSS,PRU
      GO TO 28
   30 TCSTAR=1.E30
      SS=0.0
      IF(NK.EQ.1) SS=SSTAR1-1.0
   31 SS=SS+1.0
      KSS=IFIX(SS)
      CALL FUNC(SS,F,C,TC)
      SUM=V(M,I+KSS,J)*G0
      X=1.0
      G=G0
   32 KX=IFIX(X)
      G=(RI+X-1.0)*(SS-X+1.0)/(X*(SS+TI-RI-X))*G
```

```
      SUM=SUM+V(M,I+KSS,J+KX)*G
      X=X+1.0
      IF(X.LE.SS) GO TO 32
      TCOS=TC+SUM
      IF(TCOS.GE.TCSTAR) GO TO 33
      TCSTAR=TCOS
      STAR=SS
      CTAR=C
   33 IF(SS.LT.TL) GO TO 31
      IF(TCSTAR.LE.PRU) GO TO 34
      V(L,I,J)=PRU
      GO TO 35
   34 V(L,I,J)=TCSTAR
   35 WRITE (6,17) RI,TI,RWII,AWII,TCSTAR,STAR,CTAR,V(L,I,J)
      GO TO 28
   38 V(L,I,J)=AMIN1(PRU,TCSTAR)
      WRITE(6,20) RI,TI,RWII,AWII,TCSTAR,V(L,I,J)
   28 CONTINUE
   29 CONTINUE
   99 CONTINUE
      GO TO 100
   71 AWI=CA*EP
      WRITE (6,14)
      IF(FF.GE.TES) GO TO 72
      BS(1,1)=OHI+(1.0-1.0/TL)*(FF-TES)
      GO TO 73
   72 BS(1,1)=OHI
   73 V(2,1,1)=AMIN1(CR,AWI,BS(1,1))
      IF(V(2,1,1).LT.BS(1,1).OR.FF.GE.TES) GO TO 74
      CALL SEARCH
      V(2,1,1)=AMIN1(CR,AWI,TSTAR)
      WRITE(6,17) RI,TI,CR,AWI,TSTAR,SSTAR,CSTAR,V(2,1,1)
      GO TO 100
   74 WRITE(6,19) RI,TI,CR,AWI,BS(1,1),V(2,1,1)
  100 CONTINUE
      STOP
      END
```

```
C
C
      SUBROUTINE FUNC(SS,F,C,TC)
      COMMON /DATA1/ CI,CV,CR,CA,RI,TI,EP,TL,G0
      A=1.0
      ASI=TI+SS-1.0
      G0=(ASI-RI)/ASI
    1 G0=A*G0
      IF(ASI.LE.TI) GO TO 2
      ASI=ASI-1.0
      A=(ASI-RI)/ASI
      GO TO 1
    2 IF(SS.EQ.TL) GO TO 6
      CP=CR/CA*(TI+SS)-RI
      IF(CP.LT.0.0) GO TO 8
      CP=AINT(CP)
      IF(CP.GE.SS) GO TO 4
      C=CP
      F=(CA*RI/(TI+SS)-CR)*G0
      IF(C.EQ.0.0) GO TO 5
      X=1.0
      G=G0
    3 G=(RI+X-1.0)*(SS-X+1.0)/(X*(SS+TI-RI-X))*G
      F=F+(CA*(RI+X)/(TI+SS)-CR)*G
      X=X+1.0
      IF(X.LE.C) GO TO 3
      GO TO 5
    4 C=SS
      F=CA*EP-CR
      GO TO 5
    8 C=-1.0
      F=0.0
    5 TC=(CI+SS*CV+CP*(SS*EP+TL-SS)+(TL-SS)*F)/TL
      GO TO 7
    6 TC=CI/TL+CV+CR*EP
      F=0.0
      C=0.0
    7 CONTINUE
      RETURN
      END
```

```
C
C
      SUBROUTINE SEARCH
      COMMON /DATA1/ CI,CV,CR,CA,RI,TI,EP,TL,GO
      COMMON /DATA4/ TSTAR,CSTAR,SSTAR
      COMMON /DATA5/ KF(20)
      DIMENSION P(50),P2(50),P3(50)
      TSTAR=1.E30
      EX=EP*TL
      EVPI=AMIN1(CA*EX,CR*TL)-CR*EX
      UB=AINT((EVPI-CI)/CV+0.001)
      UU=AMIN1(TL,UB)
      IF(UU.GT.1.0) GO TO 1
      CALL FUNC(1.0,F,CSTAR,TSTAR)
      SSTAR=1.0
      GO TO 16
    1 R=CR/CA
      A=R*TI-RI
      BL=AINT(A/(1.0-R)+0.001)
      BLR=AINT(-A/R)
      BB=AMAX1(0.0,BL,BLR)
      IF(BB.EQ.0.0) GO TO 5
      IF(BB.EQ.BLR) GO TO 32
      DKN=CV+EP*(CR-CA)
      IF(DKN.LT.0.0) GO TO 2
      TSTAR=(CI+CV+EP*(CR+CA*(TL-1.0)))/TL
      CSTAR=1.0
      SSTAR=1.0
      IF(BB.GE.UU) GO TO 16
    2 IF(BB.GE.UU) GO TO 4
      GO TO 5
    4 TSTAR=(CI+CV*UU+EP*(CR*UU+CA*(TL-UU)))/TL
      CSTAR=UU
      SSTAR=UU
      GO TO 16
   32 DKN=CV+CR*(EP-1.0)
      IF(DKN.LT.0.0) GO TO 33
      TSTAR=(CI+CV+CR*(TL-1.0)+CR*EP)/TL
      SSTAR=1.0
      CSTAR=-1.0
      IF(BB.GE.UU) GO TO 16
   33 IF(BB.GE.UU) GO TO 34
      GO TO 5
   34 TSTAR=(CI+CV*UU+CR*(TL-UU)+CR*UU*EP)/TL
      SSTAR=UU
      CSTAR=-1.0
      GO TO 16
    5 SPA=AINT(CA/CR+0.001)+1.0
      SON=AINT((UU-BB)/SPA+0.001)+2.0
      SM=AMAX1(SPA,SON)
      IF(SON.LT.2.0) SON=2.0
      IF(SON.GT.TL) SON=TL
      MS=IFIX(SM)
      I=1
      KF(1)=1
      KOLD=1
    6 I=I+1
      KF(I)=KF(I-1)+KOLD
      IF(KF(I).GT.MS) GO TO 7
      KOLD=KF(I-1)
      GO TO 6
    7 P(1)=BB+1.0
      NSO=IFIX(SON)
      P(NSO)=UU
      IF(NSO.EQ.2) GO TO 21
      NS1=NSO-1
      DO 9 I=2,NS1
      PK=P(I-1)+SPA+1.0
    8 PK=PK-1.0
      IC=INT(A+R*PK+0.001)
      IC1=INT(A+R*(PK-1.0)+0.001)
      IF(IC-IC1.EQ.1) GO TO 9
      GO TO 8
    9 P(I)=PK
   21 CALL FIBON(P,NSO,IS,SSI1,CST1,TCST1)
      IF(TCST1.GE.TSTAR) GO TO 10
      TSTAR=TCST1
```

```
      SSTAR=SST1
      CSTAR=CST1
   10 IF(IS.EQ.1) GO TO 24
      B1=P(IS-1)+1.0
      B2=P(IS)-1.0
      PN=B2-B1+1.0
      NP=IFIX(PN)
      I=1
      P2(1)=B1
   11 P2(I+1)=P2(I)+1.0
      IF(P2(I+1).GE.B2) GO TO 12
      I=I+1
      GO TO 11
   12 CALL FIBON(P2,NP,ISN,SST2,CST2,TCST2)
      IF(TCST2.GE.TSTAR) GO TO 13
      TSTAR=TCST2
      SSTAR=SST2
      CSTAR=CST2
   13 IF(IS.EQ.NSO) GO TO 16
   24 B3=P(IS)+1.0
      B4=P(IS+1)-1.0
      PN=B4-B3+1.0
      NP=IFIX(PN)
      I=1
      P3(1)=B3
   14 P3(I+1)=P3(I)+1.0
      IF(P3(I+1).GE.B4) GO TO 15
      I=I+1
      GO TO 14
   15 CALL FIBON(P3,NP,ISN,SST3,CST3,TCST3)
      IF(TCST3.GE.TSTAR) GO TO 16
      TSTAR=TCST3
      SSTAR=SST3
      CSTAR=CST3
   16 CONTINUE
      RETURN
      END
```

```
C
C
      SUBROUTINE FIBON(PP,NP,IS,SST,CST,TCST)
      COMMON /DATA1/ CI,CV,CR,CA,RI,TI,EP,TL,G0
      COMMON /DATA5/ KF(20)
      DIMENSION PP(100)
      I=0
    1 I=I+1
      IF(KF(I).GT.NP) GO TO 2
      GO TO 1
    2 IF(I.LE.3) GO TO 11
      CALL FUNC(PP(KF(I-1)),F,C1,TC1)
    3 CALL FUNC(PP(KF(I-2)),F,C2,TC2)
      IF(TC2.GE.TC1) GO TO 4
      I=I-1
      IF(I.LE.2) GO TO 12
      TC1=TC2
      C1=C2
      GO TO 3
    4 IF(I.EQ.3) GO TO 13
      KP=2*KF(I-2)
      IF(KP.GT.NP) KP=NP
      CALL FUNC(PP(KP),F,C3,TC3)
      IF(TC3.LE.TC1) GO TO 7
      TCST=TC1
      CST=C1
      SST=PP(KF(I-1))
      IS=KF(I-1)
      J=0
      LIM=KP-1
    5 J=J+1
      KS=KF(I-2)+J
      IF(KS.EQ.KF(I-1)) GO TO 5
      IF(KS.EQ.KP) GO TO 99
      CALL FUNC(PP(KS),F,C3,TC3)
      IF(TC3.GE.TCST) GO TO 6
      TCST=TC3
      CST=C3
      SST=PP(KS)
      IS=KS
    6 IF(KS.LT.LIM) GO TO 5
      GO TO 99
    7 LIM=MIN0(KF(I)-1,NP)
      TCST=TC3
      CST=C3
      SST=PP(KP)
      IS=KP
      J=0
    9 J=J+1
      KS=KF(I-1)+J
      IF(KS.EQ.KP) GO TO 9
      IF(KS.GE.LIM) GO TO 99
      CALL FUNC(PP(KS),F,C4,TC4)
      IF(TC4.GE.TCST) GO TO 9
      TCST=TC4
      CST=C4
      SST=PP(KS)
      IS=KS
      GO TO 9
   11 CALL FUNC(PP(1),F,C2,TC2)
      CALL FUNC(PP(2),F,C1,TC1)
      IF(TC2.LE.TC1) GO TO 12
   13 TCST=TC1
      CST=C1
      SST=PP(2)
      IS=2
      GO TO 99
   12 TCST=TC2
      CST=C2
      SST=PP(1)
      IS=1
   99 CONTINUE
      RETURN
      END
```

```
      PROGRAM BAYES3(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     METHOD 1.
C
      COMMON /DATA1/ CI,CV,CR,CA,RI,TI,EP,TL,GO
      COMMON /DATA4/ ISTAR,CSTAR,SSTAR
      DIMENSION V(2,5000),SAMSIZ(20)
      READ (5,1) NL,CI,CV,CR,CA,R,T,TL,UL
    1 FORMAT (I5,8F7.2)
      WRITE (6,2)
    2 FORMAT (2Y,7)
      WRITE(6,44)
   44 FORMAT(4X,"L    KI      KV      KR      KA      RO      TO      N",
     1"      UL")
      WRITE (6,1) NL,CI,CV,CR,CA,R,T,TL,UL
      WRITE (6,2)
      EXDEF=0.0
      SUMSAM=0.0
      I=0
    3 I=I+1
      RI=R+EXDEF
      TI=T+SUMSAM
      EP=RI/TI
      NK=NL-I+1
      TL=FLOAT(NK)*TL
      SS=TL-1.0
      IF(I.GE.NL) GO TO 6
      CALL FUNC(SS,FF,C,TC)
      TES=CV-CR*(TI-RI)/TI
      IF(FF.GE.TES) GO TO 4
      CALL SEARCH
      GO TO 5
    4 SSTAR=TL
    5 SAMSIZ(I)=SSTAR
      IF(SSTAR.GE.UL) SAMSIZ(I)=UL
      SUMSAM=SUMSAM+SAMSIZ(I)
      EXDEF=SUMSAM*R/T
      TL=TL/FLOAT(NK)
      GO TO 3
    6 NI=IFIX(SUMSAM)+1
      CS=CI/TL+CV
      LS=IFIX(TL)
      CRA=CR/CA
      ULP2=1-CV/CR
      SLP2=CV/(CA-CR)
      L=1
      DO 99 K=1,NL
      NK=NL-K+1
      WRITE (6,2)
      WRITE (6,8) NK
    8 FORMAT (2X,"STAGE NUMBER",I4)
      WRITE (6,2)
      IF(L.EQ.2) GO TO 9
      L=2
      M=1
      GO TO 10
    9 L=1
      M=2
   10 IF(K.GT.1) GO TO 91
      WRITE (6,14)
   14 FORMAT(27X,"    RI        TI        REJECT   ACCEPT ",
     1"SAMCO/LO SAMSIZ  ACCNUM  V( )")
      WRITE(6,2)
      ULP1=CRA*(1+TL/TI)
      SLP1=TL/TI*(CRA-1)+CRA
      AMAXEP=AMAX1(ULP1,ULP2)
      AMINEP=AMIN1(SLP1,SLP2)
      J=1
   98 RI=R+FLOAT(J)-1.0
      EP=RI/TI
      IF(EP.LT.AMINEP) GO TO 21
      IF(EP.GT.AMAXEP) GO TO 24
      AWI=CA*EP
      OWI=CS+CR*EP
      CALL FUNC(SS,FF,C,TC)
      TES=CV-CR*(TI-RI)/TI
      IF(FF.GE.TES) GO TO 11
```

```
      BS=OHI+(1.0-1.0/TL)*(FF-TES)
      V(L,J)=AMIN1(CR,AWI,BS)
      IF(V(L,J).LT.BS) GO TO 18
      CALL SEARCH
      V(L,J)=AMIN1(CR,AWI,TSTAR)
      WRITE (6,17) RI,TI,CR,AWI,TSTAR,SSTAR,CSTAR,V(L,J)
   17 FORMAT (2X,"OPTIMAL SAMPLE CALCULATED",8F8.3)
      GO TO 88
   18 WRITE (6,19) RI,TI,CR,AWI,BS,V(L,J)
   19 FORMAT (27X,5F8.3,16X,F8.3)
      GO TO 88
   11 V(L,J)=AMIN1(CR,AWI,OHI)
      WRITE(6,20) RI,TI,CR,AWI,OHI,V(L,J)
   20 FORMAT(2X,"100% SAMPLING",12X,5F8.3,16X,F8.3)
      GO TO 88
   21 JA=IFIX(TI*AMINEP-R+1.0)
      J=MIN0(JA,NI)
      DO 22 KL=1,J
      RII=R+FLOAT(KL)-1.0
   22 V(L,KL)=CA*RII/TI
      WRITE(6,23) RI,RII,TI
   23 FORMAT(2X,"ACCEPT WITHOUT SAMPLING IS OPTIMAL FROM RI=",
     3F8.3," TO RI=",F8.3," FOR TI=",F8.3)
      GO TO 88
   24 DO 25 KL=J,NI
   25 V(L,KL)=CR
      J=NI
      RII=R+FLOAT(NI)-1.0
      WRITE(6,26) RI,RII,TI
   26 FORMAT(2X,"REJECT WITHOUT SAMPLING IS OPTIMAL FROM RI=",
     4F8.3," TO RI=",F8.3," FOR TI=",F8.3)
   88 J=J+1
      IF(J.LE.NI) GO TO 98
      RRI=RI
      GO TO 99
   91 WRITE (6,14)
      WRITE(6,2)
      KSAMSIZ=IFIX(SAMSIZ(NK))
      NI=NI-KSAMSIZ
      TI=TI-SAMSIZ(NK)
      J=1
   71 RI=R+FLOAT(J)-1.0
      IF(RI.GE.RRI) GO TO 72
      EP=RI/TI
      AWII=FLOAT(K)*CA*EP
      RWII=FLOAT(K)*CR
      CALL FUNC(SAMSIZ(NK),F,C,TC)
      SUM=V(M,J)*GO
      X=1.0
      G=GO
   32 KX=IFIX(X)
      G=(RI+X-1.0)*(SAMSIZ(NK)-X+1.0)/(X*(SAMSIZ(NK)+TI-RI-X))*G
      SUM=SUM+V(M,J+KX)*G
      X=X+1.0
      IF(X.LE.SAMSIZ(NK)) GO TO 32
      TCOS=TC+SUM
      V(L,J)=AMIN1(AWII,RWII,TCOS)
      WRITE (6,16) RI,TI,RWII,AWII,TCOS,SAMSIZ(NK),C,V(L,J)
   16 FORMAT(27X,8F8.3)
      J=J+1
      IF(J.LE.NI) GO TO 71
      GO TO 99
   72 RWII=FLOAT(K)*CR
      DO 75 KL=J,NI
   75 V(L,KL)=RWII
      RII=R+FLOAT(NI)-1.0
      WRITE(6,26) RI,RII,TI
   99 CONTINUE
      STOP
      END
```

```
      PROGRAM BAYES4(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     METHOD 2A.
C
      COMMON /DATA1/ CI,CV,CR,CA,RI,TI,EP,TL,GO
      COMMON /DATA4/ TSTAR,CSTAR,SSTAR
      DIMENSION V(2,5000)
      READ (5,1) NL,CI,CV,CR,CA,R,T,TL,UL
    1 FORMAT (I5,8F7.2)
      WRITE (6,2)
    2 FORMAT (2X,/)
      WRITE(6,44)
   44 FORMAT(4X,"L    KI      KV      KR      KA      R0      T0      N",
     1"      UL")
      WRITE (6,1) NL,CI,CV,CR,CA,R,T,TL,UL
      WRITE (6,2)
      RI=R
      TI=T
      EP=RI/TI
      TL=FLOAT(NL)*TL
      SS=TL-1.0
      CALL FUNC(SS,FF,C,TC)
      TES=CV-CR*(TI-RI)/TI
      IF(FF.GE.TES) GO TO 4
      CALL SEARCH
      GO TO 5
    4 SSTAR=TL
    5 SAMSIZ=SSTAR
      IF(SSTAR.GE.UL) SAMSIZ=UL
      TL=TL/FLOAT(NL)
    6 NI=IFIX(SAMSIZ)*(NL-1)+1
      TI=TI+SAMSIZ*(FLOAT(NL)-1.0)
      CS=CI/TL+CV
      LS=IFIX(TL)
      CRA=CR/CA
      ULP2=1-CV/CR
      SLP2=CV/(CA-CR)
      L=1
      DO 99 K=1,NL
      NK=NL-K+1
      WRITE (6,2)
      WRITE (6,8) NK
    8 FORMAT (2X,"STAGE NUMBER:",I4)
      WRITE (6,2)
      IF(L.EQ.2) GO TO 9
      L=2
      M=1
      GO TO 10
    9 L=1
      M=2
   10 IF(K.GT.1) GO TO 91
      WRITE (6,14)
   14 FORMAT(27X,"    RI        TI        REJECT   ACCEPT ",
     1"SAMCO/L9  SAMSIZ   ACCNUM   V( )")
      WRITE(6,2)
      ULP1=CRA*(1+TL/TI)
      SLP1=TL/TI*(CRA-1)+CRA
      AMAXEP=AMAX1(ULP1,ULP2)
      AMINEP=AMIN1(SLP1,SLP2)
      J=1
   98 RI=R+FLOAT(J)-1.0
      EP=RI/TI
      IF(EP.LT.AMINEP) GO TO 21
      IF(EP.GT.AMAXEP) GO TO 24
      AWI=CA*EP
      OHI=CS+CR*EP
      CALL FUNC(SS,FF,G,TC)
      TES=CV-CR*(TI-RI)/TI
      IF(FF.GE.TES) GO TO 11
      BS=OHI+(1.0-1.0/TL)*(FF-TES)
      V(L,J)=AMIN1(CR,AWI,BS)
      IF(V(L,J).LT.BS) GO TO 18
      CALL SEARCH
      V(L,J)=AMIN1(CR,AWI,TSTAR)
      WRITE (6,17) RI,TI,CR,AWI,TSTAR,SSTAR,CSTAR,V(L,J)
   17 FORMAT (2X,"OPTIMAL SAMPLE CALCULATED",8F8.3)
      GO TO 88
```

```
   18 WRITE (6,19) RI,TI,CR,AWI,BS,V(L,J)
   19 FORMAT (27X,5F8.3,16X,F8.3)
      GO TO 88
   11 V(L,J)=AMIN1(CR,AWI,OHI)
      WRITE(6,20) RI,TI,CR,AWI,OHI,V(L,J)
   20 FORMAT(2X,"100% SAMPLING",12X,5F8.3,16X,F8.3)
      GO TO 88
   21 JA=IFIX(TI*AMINEP-R+1.0)
      J=MIN0(JA,NI)
      DO 22 KL=1,J
      RII=R+FLOAT(KL)-1.0
   22 V(L,KL)=CA*RII/TI
      WRITE(6,23) RI,RII,TI
   23 FORMAT(2X,"ACCEPT WITHOUT SAMPLING IS OPTIMAL FROM RI=",
     3F8.3," TO RI=",F8.3," FOR TI=",F8.3)
      GO TO 88
   24 DO 25 KL=J,NI
   25 V(L,KL)=CR
      J=NI
      RII=R+FLOAT(NI)-1.0
      WRITE(6,26) RI,RII,TI
   26 FORMAT(2X,"REJECT WITHOUT SAMPLING IS OPTIMAL FROM RI=",
     4F8.3," TO RI=",F8.3," FOR TI=",F8.3)
   88 J=J+1
      IF(J.LE.NI) GO TO 98
      RRI=RI
      GO TO 99
   91 WRITE (6,14)
      WRITE(6,2)
      KSAMSIZ=IFIX(SAMSIZ)
      NI=NI-KSAMSIZ
      TI=TI-SAMSIZ
      J=1
   71 RI=R+FLOAT(J)-1.0
      IF(RI.GE.RRI) GO TO 72
      EP=RI/TI
      AWII=FLOAT(K)*CA*EP
      RWII=FLOAT(K)*CR
      CALL FUNC(SAMSIZ,F,C,TC)
      SUM=V(M,J)*GO
      X=1.0
      G=GO
   32 KX=IFIX(X)
      G=(RI+X-1.0)*(SAMSIZ-X+1.0)/(X*(SAMSIZ+TI-RI-X))*G
      SUM=SUM+V(M,J+KX)*G
      X=X+1.0
      IF(X.LE.SAMSIZ) GO TO 32
      TCOS=TC+SUM
      V(L,J)=AMIN1(AWII,RWII,TCOS)
      WRITE (6,16) RI,TI,RWII,AWII,TCOS,SAMSIZ,C,V(L,J)
   16 FORMAT(27X,6F8.3)
      J=J+1
      IF(J.LE.NI) GO TO 71
      GO TO 99
   72 RWII=FLOAT(K)*CR
      DO 75 KL=J,NI
   75 V(L,KL)=RWII
      RII=R+FLOAT(NI)-1.0
      WRITE(6,26) RI,RII,TI
   99 CONTINUE
      STOP
      END
```

```
      PROGRAM BAYES5(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     METHOD 28.
C
C
C
      COMMON /DATA1/ CI,CV,CR,CA,RI,TI,EP,TL,GO
      COMMON /DATA4/ TSTAR,CSTAR,SSTAR
      DIMENSION V(2,5000)
      READ (5,1) NL,CI,CV,CR,CA,R,T,TL,UL
    1 FORMAT (I5,8F7.2)
      WRITE (6,2)
    2 FORMAT (2X,/)
      WRITE(6,44)
   44 FORMAT(4X,"L    KI      KV      KR      KA      R0      T0      N",
     1"      UL")
      WRITE (6,1) NL,CI,CV,CR,CA,R,T,TL,UL
      WRITE (6,2)
      RI=R
      TI=T
      EP=RI/TI
      SS=TL-1.0
      CALL FUNC(SS,FF,C,TC)
      TES=CV-CR*(TI-RI)/TI
      IF(FF.GE.TES) GO TO 4
      CALL SEARCH
      GO TO 5
    4 SSTAR=TL
    5 SAMSIZ=SSTAR
      IF(SSTAR.GE.UL) SAMSIZ=UL
    6 NI=IFIX(SAMSIZ)*(NL-1)+1
      TI=TI+SAMSIZ*(FLOAT(NL)-1.0)
      CS=CI/TL+CV
      LS=IFIX(TL)
      CRA=CR/CA
      ULP2=1-CV/CR
      SLP2=CV/(CA-CR)
      L=1
      DO 99 K=1,NL
      NK=NL-K+1
      WRITE (6,2)
      WRITE (6,8) NK
    8 FORMAT (2X,"STAGE NUMBER:",I4)
      WRITE (6,2)
      IF(L.EQ.2) GO TO 9
      L=2
      M=1
      GO TO 10
    9 L=1
      M=2
   10 IF(K.GT.1) GO TO 91
      WRITE (6,14)
   14 FORMAT(27X,"   RI        TI      REJECT  ACCEPT ",
     1"SAMCO/LB SAMSIZ  ACCNUM  V( )")
      WRITE(6,2)
      ULP1=CRA*(1+TL/TI)
      SLP1=TL/TI*(CRA-1)+CRA
      AMAXEP=AMAX1(ULP1,ULP2)
      AMINEP=AMIN1(SLP1,SLP2)
      J=1
   98 RI=R+FLOAT(J)-1.0
      EP=RI/TI
      IF(EP.LT.AMINEP) GO TO 21
      IF(EP.GT.AMAXEP) GO TO 24
      AWI=CA*EP
      OHI=CS+CR*EP
      CALL FUNC(SS,FF,C,TC)
      TES=CV-CR*(TI-RI)/TI
      IF(FF.GE.TES) GO TO 11
      BS=OHI+(1.0-1.0/TL)*(FF-TES)
      V(L,J)=AMIN1(CR,AWI,BS)
      IF(V(L,J).LT.BS) GO TO 18
      CALL SEARCH
      V(L,J)=AMIN1(CR,AWI,TSTAR)
      WRITE (6,17) RI,TI,CR,AWI,TSTAR,SSTAR,CSTAR,V(L,J)
   17 FORMAT (2X,"OPTIMAL SAMPLE CALCULATED",8F8.3)
```

```
      GO TO 88
   18 WRITE (6,19) RI,TI,CR,AWI,BS,V(L,J)
   19 FORMAT (27X,5F8.3,16X,F8.3)
      GO TO 88
   11 V(L,J)=AMIN1(CR,AWI,OHI)
      WRITE(6,20) RI,TI,CR,AWI,OHI,V(L,J)
   20 FORMAT(2X,"100% SAMPLING",12X,5F8.3,16X,F8.3)
      GO TO 88
   21 JA=IFIX(TI*AMINEP-R+1.0)
      J=MIN0(JA,NI)
      DO 22 KL=1,J
      RII=R+FLOAT(KL)-1.0
   22 V(L,KL)=CA*RII/TI
      WRITE(6,23) RI,RII,TI
   23 FORMAT(2X,"ACCEPT WITHOUT SAMPLING IS OPTIMAL FROM RI=",
     3F8.3," TO RI=",F8.3," FOR TI=",F8.3)
      GO TO 88
   24 DO 25 KL=J,NI
   25 V(L,KL)=CR
      J=NI
      RII=R+FLOAT(NI)-1.0
      WRITE(6,26) RI,RII,TI
   26 FORMAT(2X,"REJECT WITHOUT SAMPLING IS OPTIMAL FROM RI=",
     4F8.3," TO RI=",F8.3," FOR TI=",F8.3)
   88 J=J+1
      IF(J.LE.NI) GO TO 98
      RRI=RI
      GO TO 99
   91 WRITE (6,14)
      WRITE(6,2)
      KSAMSIZ=IFIX(SAMSIZ)
      NI=NI-KSAMSIZ
      TI=TI-SAMSIZ
      J=1
   71 RI=R+FLOAT(J)-1.0
      IF(RI.GE.RRI) GO TO 72
      EP=RI/TI
      AWII=FLOAT(K)*CA*EP
      RWII=FLOAT(K)*CR
      CALL FUNC(SAMSIZ,F,C,TC)
      SUM=V(M,J)*G0
      X=1.0
      G=G0
   32 KX=IFIX(X)
      G=(RI+X-1.0)*(SAMSIZ-X+1.0)/(X*(SAMSIZ+TI-RI-X))*G
      SUM=SUM+V(M,J+KX)*G
      X=X+1.0
      IF(X.LE.SAMSIZ) GO TO 32
      TCOS=TC+SUM
      V(L,J)=AMIN1(AWII,RWII,TCOS)
      WRITE (6,16) RI,TI,RWII,AWII,TCOS,SAMSIZ,C,V(L,J)
   16 FORMAT(27X,8F8.3)
      J=J+1
      IF(J.LE.NI) GO TO 71
      GO TO 99
   72 RWII=FLOAT(K)*CR
      DO 75 KL=J,NI
   75 V(L,KL)=RWII
      RII=R+FLOAT(NI)-1.0
      WRITE(6,26) RI,RII,TI
   99 CONTINUE
      STOP
      END
```