AN ABSTRACT OF THE THESIS OF

BONG JIN YUM for the degree of MASTER OF SCIENCE

in INDUSTRIAL ENGINEERING presented on June 10, 1977

Title: OPTIMAL ALLOCATION OF SCREENING INSPECTION EFFORT FOR NONSERIAL PRODUCTION SYSTEMS

Abstract approved: Redacted for Privacy

Dr. Edward McDowell

The economic aspects of quality control have received wide concern since its developmental stages, however, they have not been sufficiently explored. Moreover, as the need for complex products of high precision and reliability increases, most industrial firms have experienced some difficulties in meeting that need at reasonable costs with their limited capabilities.

Realizing that one of the most efficient ways of overcoming such difficulties is to properly allocate their inspection efforts within a production process, a mathematical model has been developed for determining the optimal location of screening inspection stations within a nonserial production process. Especially, the effect of imperfect inspection on the probabilistic behaviors of good and defective items is explicitly considered.

The model is formulated as a constrained nonlinear zero-one integer program, and it is proved flexible enough to reflect various limitations and requirements existing in general production-inspec-

tion systems.

As solution techniques, Lawler and Bell's algorithm and Glover's enumeration scheme are proposed with proper fathoming tests obtained by identifying the unique structure of the problem. Glover's scheme is modified to avoid complicated bookkeeping procedures, and FORTRAN programs are developed for both algorithms. The computational efficiencies of the two algorithms are compared with each other by solving randomly generated test problems, and it is shown that Glover's enumeration scheme is more efficient than Lawler and Bell's algorithm under the given conditions for generating test problems. However, it is realized that Glover's scheme is not quite efficient as the problem size increases. Thus, additional research is desired for accelerating enumeration in Glover's scheme.

In addition, a nonserial production-inspection system is analyzed by performing some sensitivity analyses showing that the model is adaptive to varying situations.

Optimal Allocation of Screening Inspection Effort
For Nonserial Production Systems

by

Bong Jin Yum

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed June 10, 1977

Commencement June 1978

APPROVED:

Redacted for Privacy

Assistant Professor of Industrial Engineering
in charge of major

Redacted for Privacy

Chairman of the Department of Industrial and General Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented June 10, 1977

Typed by Larky Hansen for Bong Jin Yum

ACKNOWLEDGEMENT

The author acknowledges special appreciation to Dr. Edward McDowell, his major advisor, for his guidance and assistance in the preparation of this study.

Acknowledgement is also given to Dr. James L. Riggs, Chairman of the Department of Industrial and General Engineering, for his kind consideration during the course of study.

The author wishes to express his gratitude to his parents for their help and encouragement.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

OPTIMAL ALLOCATION OF SCREENING INSPECTION EFFORT FOR NONSERIAL PRODUCTION SYSTEMS

I. INTRODUCTION

The Problem

Juran (1974) has estimated that the total quality costs incurred to maintain the desired level of product quality in most industrial firms lies between 5 and 10% of sales, and in some cases it can run to over 25%. Thus, industrial quality control groups have been faced with internal pressures for reducing these enormous costs. On the other hand, they have been faced with external forces calling for the improvement of the product quality at the same time. Moreover, due to increasing needs for more complex products, greater precision, and high reliability, the cost of achieving product fitness for use has increased. In other words, those pressures from both internal as well as external sources have become higher and higher. Thus, there has been an urgent need to settle the conflict of those two pressure groups by providing adequate decision criteria for minimizing the quality costs while achieving the desired quality level under given conditions.

One of the most efficient ways for solving this problem is to properly allocate inspection effort within the production system considered. For this "proper" allocation of inspection effort, several rules of thumb have prevailed in industrial firms. One generally accepted rule is to calculate the ratio of the expected internal failure costs to the inspection labor costs at each potential inspection station. This rule is applied in such a manner that inspection is performed only when the ratio is greater than one at a particular inspection station. This rule is fairly simple, however, it ignores the fact that in any sequential production systems the decision made at a given inspection station has a significant influence not only on the costs incurred at that station, but also on the product quality at all subsequent operations, and, as a result, affects the decisions at all subsequent inspection stations. A truly optimal solution cannot be obtained by treating each inspection station independently using this kind of rule. A mathematical model needs to be established to reflect these interactions explicitly.

Several important aspects to be considered for building a mathematical model include: the nature of the product and the production systems; inspection policy and inspection accuracy; and the nature of quality costs.

The effects of each of these factors on the problem are described in detail in the following sections.

The Nature of the Product and Physical Structures of Production Systems

Products take on multiple forms while progressing through the production process, and may exhibit different characteristics when submitted to inspection. Juran (1974) describes the nature of the product as shown in Table 1-1.

Probabilistic behaviors of good or bad items in the production

Table 1-1. The Nature and Forms of Product

Nature of Product	Forms	Descriptions
Single Elements	Discrete units	Separate entities
	Specimens from a coalesced mass.	Samples from batches or continuous processes
Lots	Collection of discrete units	Obvious boundaries of batches or arbitrary amount of production
	Coalesced mass	

process as well as product acceptance policy at each inspection station largely depend on the nature of the product to be submitted to inspection. In this research, it is assumed that the product takes the form of discrete units and it is submitted as a single element to inspection.

Nowadays many products are built through sequential production processes progressing from basic materials to components, units, equipments, and systems. Sequential production processes can be classified into two groups: serial (single-channel) processes; and nonserial processes. Because nonserial structures appear more commonly in our modern process industries this research is concentrated upon nonserial production systems which include serial or sequential production processes as special cases.

In a nonserial production process complicated interactions may arise between operations in the form of converging, diverging, feedforward, and feedback structures. Converging branches occur when the outputs from two or more different operations are combined into an operation; diverging branches occur when two or more outputs from an operation become the inputs to different operations; feedforward loops consist of a diverging branch from an operation, which converges at a later operation in the process; and feedback loops consist of a diverging branch from an operation, which converges at an earlier operation in the process. These complicated structures make it difficult to describe the interactions between operations in a nonserial production system. A special consideration should be given to describe them explicitly and conveniently.

Another important aspect of nonserial processes is the flow of materials between operations. The ratio of the input to the output in each operation needs not be unity. However, in most cases product is built gradually by assembling parts, components, units, etc., and therefore, this ratio is some positive integer.

Besides, a nonserial production process may have several different raw material suppliers as well as different final operations in which different types of final products are produced.

Typical examples for a serial and a nonserial production process are illustrated in Figure 1-1 and Figure 1-2, respectively.

Inspection Policy and Inspection Accuracy

The term "inspection" used here implies "acceptance inspection", which consists of evaluating the quality of some characteristics of the product in order to judge conformance to specification.

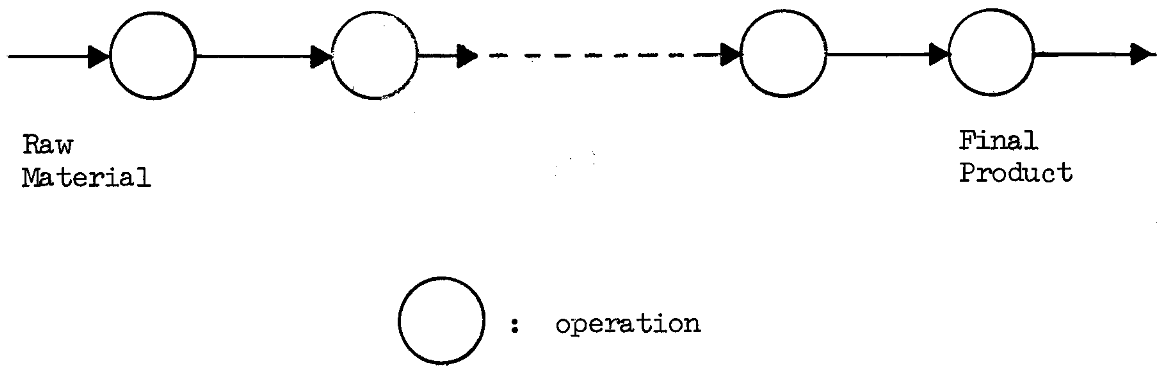

Figure 1-1. A Typical Serial Production Process

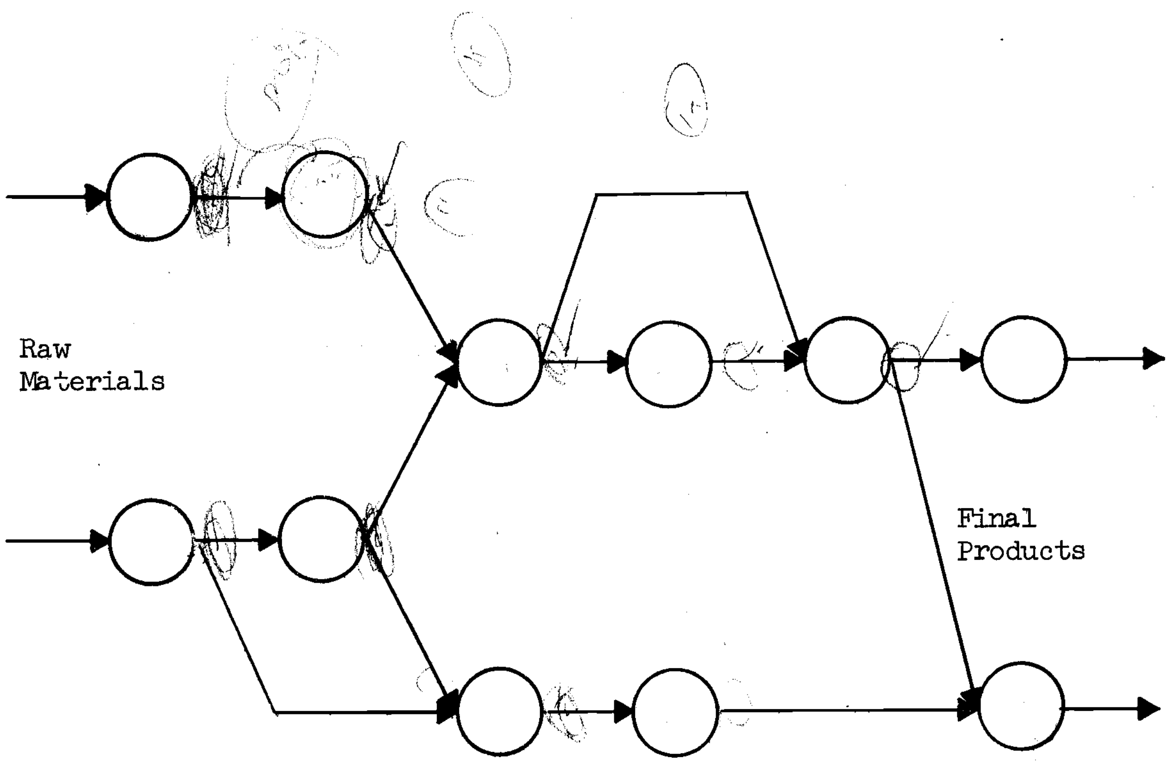

Figure 1-2. A Typical Nonserial Production Process

According to the judgment made the product may be accepted for further processing or final use, and on the other hand, it may be scrapped, repaired, and so forth. This product acceptance can be performed by two major types of inspection. One is screening or 100% inspection and the other is sampling inspection. Although sampling inspection is usually the more economical procedure, there exist several situations in which 100% inspection is desired as well as efficient. Lindsay and Bishop (1964) illustrated production processes in which existing raw material, equipment, and production techniques are frequently incapable of yielding finished product of marketable quality. They recommended that in this situation acceptable quality can only be achieved by inspecting each item produced and removing from production those items found to be defective. The Statistical Quality Control Handbook published by Western Electric Co., Inc. (1958) also illustrated several situations where 100% inspection is quite common. They include the situation where extremely high level of quality should be attained because a defective is critical to the safety or health of human beings, and also include the situation where the amount of production is fairly small and sampling is not practical. In addition to these situations, there is another important case in which 100% inspection is desired. Suppose that in a production process machining costs at each operation are so expensive that undetected defective items incur a large amount of unnecessary machining costs at successive operations. In this case, it is frequently economical to detect a defective item as early as possible by sorting.

In the above situations 100% inspection serves to improve the product quality by eliminating defective items directly from the production processes considered.

Another important aspect of inspection is the policy for disposing of detected defective product. When defects are found there are numerous actions that may be taken. A detected defective item may be scrapped or repaired according to the policy adopted. If scrapped, it loses cumulative manufacturing costs with or without some salvage values. If repaired, it may be sent to previous operations or to repair facilities incurring additional costs. The disposal policy also affects the probabilistic behavior of product processed. Thus, disposition of detected defective items is closely related with internal as well as external economics of industrial firms.

Another important factor which is closely related with the probabilistic behaviors of good and defective items in the process is inspection accuracy. If each inspection is assumed to be perfect the quality level of an item after each inspection will be perfect, however, this is not the case in actual situations.

Almost every inspection task performed by human beings is subject to significant errors, and Juran (1974) cassifies inspector errors into three categories: technique errors due to lack of skill or know-how; inadvertent errors caused by inherent human capabilities; and willful errors characterized by the fact that the inspector knows he is committing the error and that he intends to keep it up. It is widely accepted that the inspectors cannot find all the defects present due to those errors. Therefore, a good item may be classified as bad while a bad item may be classified as good by an inspector. Related conditional probabilities for those undesirable errors can be measured, and can be applied to describing the probabilistic behaviors of an item in the process.

The Nature of Quality Costs

Groocock (1974) defined quality costs as "costs associated with making defective product". In other words, quality costs include all costs which would disappear if there were no defects. In general, they have been classified into the following four categories:

1. Internal failure costs incurred by scrapping or reworking detected defective items in the process. They may include the loss of manufacturing costs, or additional costs for the rework.
2. External failure costs incurred by defective items shipped to the customer. They may include the costs for field service, returned material handling, etc. Other intangible costs incurred by losing goodwill can be included in this category with proper estimation.
3. Appraisal costs incurred by identifying defectives and separating them from conforming items during the whole process. They may include the costs for inspection

labor, maintenance of test equipment, materials consumed, etc.

4. Prevention costs incurred to maintain the overall quality program. They may include the costs for inspection planning, design of test equipment, personal training, etc.

According to Groocock (1974) the major advantage of categorizing quality costs in this way is that it makes the accounting easy by determining quantitatively and completely the groups responsible for particular quality costs.

The ratios of the above costs vary widely among industrial firms, however, Kirkpatrick (1970) mentioned that typically the first three take a major portion of the total quality costs while prevention costs are usually under 10% of all quality costs. Moreover, the prevention costs are less variable than any other quality costs. Thus, if the cost criterion per unit of product produced is adopted, the effect of the prevention costs could be fixed or ignored.

Another important aspect of the problem is the structure of each type of quality costs. Although there is no unified agreement on that matter it is common that the appraisal and the internal failure costs are assumed to be linear with respect to the unit of product insepcted. However, sometimes it is difficult to estimate the external failure costs because some intangible factor associated with the loss of goodwill can be included. One of the alternatives to overcome such difficulty is to analyze the problem with varying external failure costs.

The Objectives

Thus far, the current need for an extensive analysis of the problem has been discussed along with an introduction of the essential aspects of the problem. The literature review indicates that fairly extensive contributions have been made for optimal allocation of inspection effort within a production process, however, various important aspects of the problem have not been fully explored or reflected in each of the contributions. Most have concentrated on serial production systems where perfect inspection is assumed, and have not explicitly considered various other limitations and requirements existing in the production-inspection systems. Therefore, there is a need to expand previous models in order to overcome such deficiencies.

The primary objective of this research is to develop a mathematical model for optimal allocation of screening inspection effort for nonserial production systems where imperfect inspection is assumed. It is desired that the model be flexible enough to reflect essential aspects, limitations, and requirements of the problem explicitly.

The secondary objective is to provide a solution technique for the developed model through investigating special structures of the problem. Based upon the solution technique provided, the nature of nonserial production-inspection system is analyzed by evaluating the effects of various factors on the optimal decision policy.

II. REVIEW OF LITERATURE

Fairly extensive contributions have been made for determining optimal locations of inspection stations in sequential production systems. They have concerned themselves with various types of formulations with special features, however, they can be classified into several broad categories as shown in Figure 2-1 according to the following criteria:

1. Type of production systems considered:
    a) Serial systems
    b) Nonserial systems
2. Inspection policy adopted:
    a) Screening inspection
    b) Sampling inspection
3. Inspection accuracy assumed:
    a) Perfect inspection
    b) Imperfect inspection

According to the above classifications, this research is related with Group VI, however, a survey of other groups may be helpful to understand various aspects of production-inspection systems as well as to identify the place which this research holds in this area.

Most of previous works have concentrated on the problems arising in serial production systems where perfect screening inspection is performed, i.e., Group I in Figure 2-1. They include Lindsay and Bishop (1964), White (1966, 1969), Pruzan and Jackson (1967), Lindsay (1967), and Trippi (1974, 1975). Among them Lindsay and

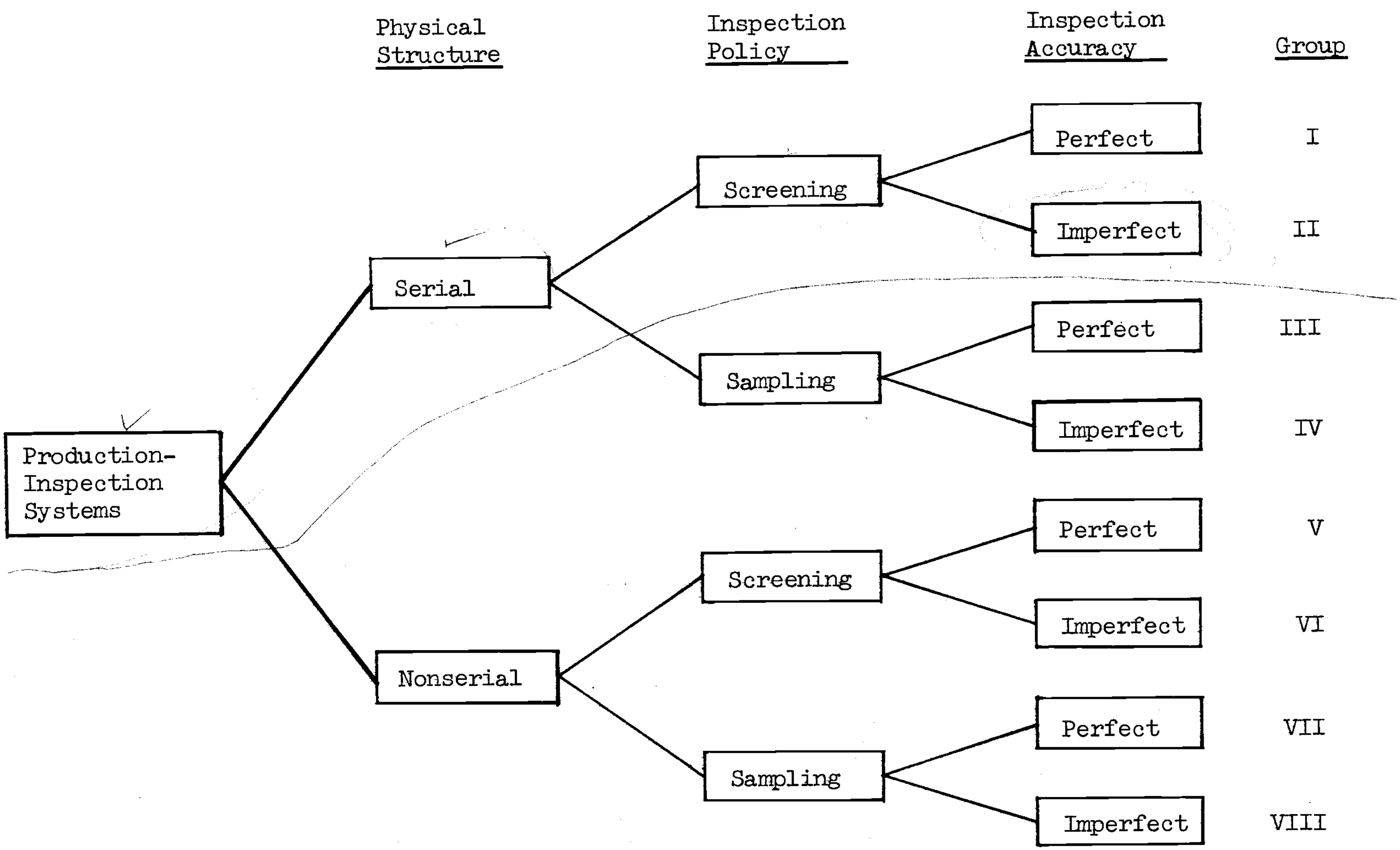

Figure 2-1. Classification of Production-Inspection Systems

Bishop are thought to be the pioneers in this area. They formulated the problem as a dynamic program in which the required quality of the final product is considered. They also established the theorem that all-or-none inspection is optimal under fairly general conditions. This theorem was proved again by White (1966) extensively, and it has formed the foundation of other contributions. White (1969) and Trippi (1974) formulated the problem as shortest route models in which defect types are classified into repairable and non-repairable ones, and also considered the availability of inspection effort. In their network model, a node represents an operation, and the length of the arc from one node to the other represents the related quality costs. They obtained an optimal inspection policy by identifying the lowest-cost route in the network. The unique feature of Pruzan and Jackson is that they presented a possibility of automated production-inspection process through on-line computation. In order to develop an optimal policy, their model uses the information about where the most recent inspection was performed and how many non-defective items are remaining after that inspection. Their model is adaptive in the sense that the optimal location of the next inspection station is determined based upon the above available information. Lindsay (1967) developed a model for the case in which each operation may generate multiple defects. Besides, it has been proved by Trippi (1975) that the problem can be formulated as a warehouse location problem which seeks an assignment of customers to warehouses to minimize the cost of distribution. His formulation of the problem is to:

$$\text{Minimize} \sum_{j=1}^{n} f_j\, y_j + \sum_{j=1}^{n} \sum_{i=1}^{m} v_{ij}\, x_{ij}$$

$$\text{Subject to} \sum_{j=1}^{n} x_{ij} = 1 \quad \text{for all } i$$

$$0 \leqslant x_{ij} \leqslant y_j \leqslant 1 \qquad x_{ij},\ y_j \text{ integer for all } i, j$$

where m = total number of defect types

n = total number of potential inspection stations

f_j = the fixed costs to maintain the j^{th} inspection station

v_{ij} = the expected unit cost of inspecting for defect type i at inspection station j.

= sum of the unit cost of repair and the unit inspection cost for defect type i at point j.

In the above cases, the assumption of perfect inspection simplifies the problem in describing the probabilistic behaviors of good or bad items because the outputs from an inspection station is perfect regardless of the incoming quality of inputs to that station. Therefore, it is possible to describe the current quality of products by knowing only the station where inspection is last performed. However, if inspection is imperfect, it is necessary to know the whole history of inspection to describe the current quality of products, and therefore, the problem becomes complicated and difficult to solve. Eppen and Hurst (1974) considered serial production-inspection systems in which imperfect screening inspection is performed, i.e., Group II. They considered two types of inspection errors as discussed in Chapter I, and described the probabilistic behaviors of good and bad items by finding related conditional probabilities from Bayes' rule. Thus, their contribution for imperfect inspection marked a step forward in this area.

The researches in Group III include Beightler and Mitten (1964), Fruehwirth (1970), Woo and Metcalfe (1972), and Dietrich and Sanders (1974). They developed systematic methods of determining optimal sampling policy throughout serial production processes under the assumption of perfect inspection. A problem in this group is more difficult to be analyzed than a problem in Group I or II because at each potential inspection station optimal sample size as well as optimal acceptance number should be determined at the same time.

For nonserial production systems, only Britney (1972) has developed a mathematical model formulated as an unconstrained zero-one nonlinear integer programming problem under the assumptions of multilinear (quasi-concave) quality cost structures and perfect screening inspection. However, it seems unreasonable to treat this production-inspection system as an unconstrained problem. Most industries are faced with such constraints as desired outgoing quality levels, manpower shortage, physical difficulties in allocating inspection at some station, etc. According to his model any final outgoing quality levels are theoretically allowed as long as they minimize the total quality costs. This contradicts that any sound quality program should keep the balance of total quality costs and the quality of final product.

Moreover, his assumption of perfect inspection may be unrealistic in some cases because it has been known that human inspection is inevitably subject to various types of errors, and therefore, any measure for outgoing quality levels would be meaningless without considering the effect of those errors (Case et al., 1975). As mentioned earlier this research aims to make up for such deficiencies in Britney's model by considering various limitations and requirements in nonserial production systems under the assumption of imperfect inspection. Therefore, this research can be included in Group VI.

It seems that there have been no extensive contributions pertaining to Group VII or VIII in which sampling inspection is considered for nonserial production-inspection systems. Although the problem in these groups is not considered in this research, it may be a useful area of future study.

III. MODEL FORMULATION

General

Consider a nonserial production system with n operations and n potential inspection stations immediately after each operation as shown in Figure 3-1. A circle and a square are used to represent an operation and an inspection station, respectively. There exist m raw material suppliers and they can be considered as dummy operations. For the convenience of the model formulation and computer programs several rules are established for numbering each operation and potential inspection station. They include:

Rule 1: Raw material suppliers are numbered prior to any other actual operations.

Rule 2: The number assigned to an operation is always greater than that of any other previous ones. The inputs to an operation always comes from operations with smaller numbers.

Rule 3: A potential inspection station is assigned the same number as that of the related operation.

In addition, only one final operation is assumed, however, this assumption can be modified easily in order to allow several different final operations in the system. That modification will be discussed later.

At the j^{th} operation, e_j is defined as a probability that an item processed at the j^{th} operation attains a defect regardless of the previous defects incurred, and these e_j's are assumed to be known. For dummy operations, e_j can be interpreted as an Average Outgoing Quality Level (AOQL) for the j^{th} raw material.

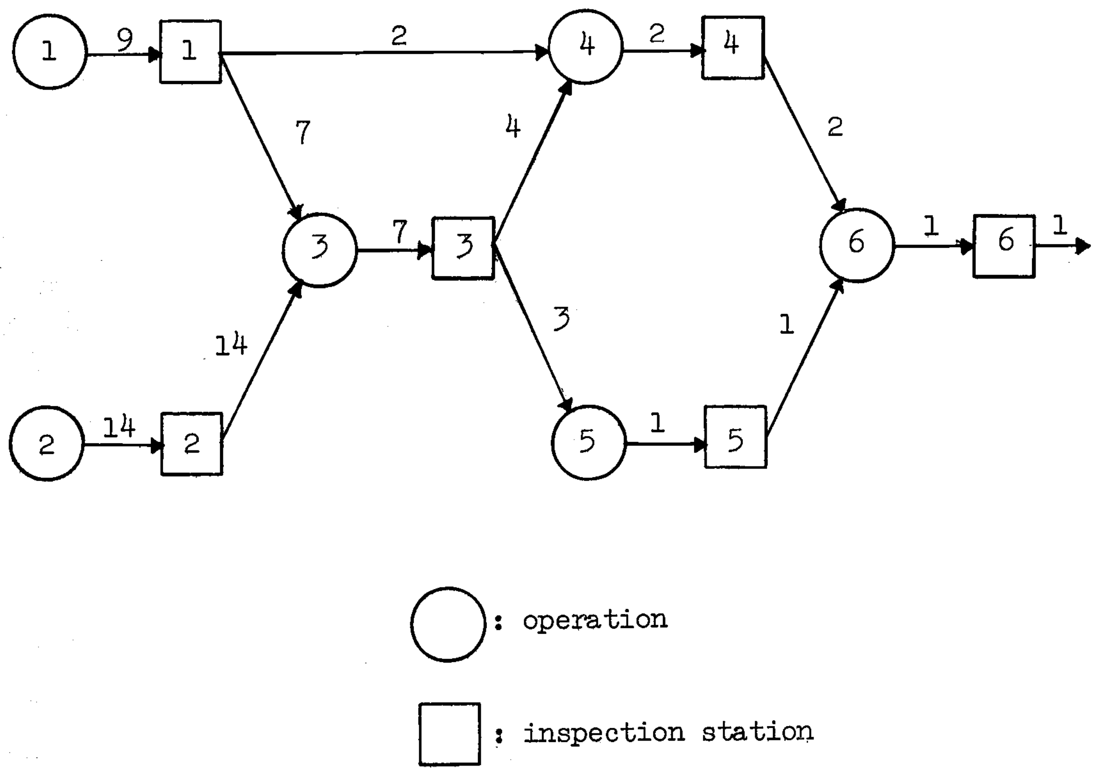

Figure 3-1. Physical Structure of Nonserial Production-Inspection System

At each potential inspection station all-or-none screening inspection is performed, and an overall screening program can be expressed as a policy vector d which is an ordered n-tuple (d_n, d_{n-1}, ...,d_1). Each d_j is a decision variable at the j^{th} inspection station, and is either 0 or 1. If inspection is performed a value of 1 is assigned, and if not the value 0 is given. For the convenience of formulation the other vector, d' is defined such that $d'_j = 1 - d_j$ for j = 1,2,......,n.

Each inspection is subject to two types of inspection errors.

A Type I error occurs when a good item is classified as a defective one while Type II error occurs when a defective item is classified as a good one by an inspector. Related conditional probabilities are assumed to be known at each inspection station.[1]

Defect types are not classified, and an item which has at least one type of defect is scrapped by detection with some salvage values. Hence, a detected defective item incurs internal failure costs by losing part of cumulative manufacturing costs which have been added. In order to maintain the equilibrium conditions of the production process, it is assumed that a detected item is replaced by a good one immediately.

Appraisal costs incurred by direct labor, equipment used, service, etc. are assumed to be linear per unit product inspected. Any defective item which escapes from detection and reaches the customer causes external failure costs regardless of the type of defects discovered. External failure costs per unit of product discovered are assumed to be known on the average. It is usual for estimating the external failure costs to include an intangible factor associated with the loss of goodwill. Thus, sometimes it is difficult to estimate them precisely. Such difficulty will be resolved by performing some sensitivity analysis with respect to external failure costs as shown in Chapter V. The effect of the prevention costs is ignored based upon the fact mentioned in Chapter I.

---

[1]For further discussion of how to measure those probabilities, see Juran and Gryna (1970), pp. 321-323.

As discussed earlier there exist various types of requirements and limitations in most production systems. The most common one is the requirement of final product quality which is represented by AOQL. That requirement is considered in the model as a constraint.

Therefore, the problem can be defined as finding an optimal policy vector d which minimizes the sum of the costs mentioned above while satisfying the requirement of final AOQL.

Material Flow Rates

It can be easily obtained from the nature of a production process in equilibrium conditions how many units of intermediate product are required between connected operations in order to produce one unit of final product.

Let r_{ij} = material flow rate from the i^{th} inspection station to the j^{th} operation to achieve one unit of final product. $(i \neq j)$

= material flow rate from the i^{th} operation to the i^{th} inspection station $(i = j)$.

Each r_{ij} is expressed on each path in Figure 3-1. For example, in order to produce one unit of final product, four units of intermediate product are required between the third inspection stations and the fourth operation, and fourteen units of the second raw materials should be supplied to produce one unit of final product. Note that

$$r_{ii} = \sum_{j=i+1}^{n} r_{ij} \quad \text{for all} \quad i = 1,2,\ldots,n$$

because detected items are assumed to be replaced by good ones immediately.

In assembly processes, a product is usually built gradually by assembling parts, components, units, etc. Therefore, the ratio of the inputs to the outputs at an operation, i.e., r_{ij}/r_{jj} is assumed to be some positive integer.

Overall material flows in the system can be expressed by a flow-rate matrix R_{nxn} (r_{ij}). Figure 3-2 is the R matrix corresponding to the system in Figure 3-1. It is always a triangular matrix which implies that there are no feedback flows in the system. The flow-rate matrix can also be used to identify connections between operations and inspection stations. The 0's in the matrix imply that there are no paths between corresponding operations. Thus, each row has at least two non-zero elements except the last row, and each column

$$R= \begin{bmatrix} 9 & 0 & 7 & 2 & 0 & 0 \\ 0 & 14 & 14 & 0 & 0 & 0 \\ 0 & 0 & 7 & 4 & 3 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 3-2. An Example of Flow-Rate Matrix has at least two non-zero elements except the columns for raw material suppliers.

Cumulative Manufacturing Costs

Let c_j be the machining costs per unit of product produced at the j^{th} operation. For each dummy operation, c_j is the purchasing costs per unit of the j^{th} raw material. Let s_j be the cumulative manufacturing costs per unit of intermediate product immediately after the j^{th} operation, and then, s_j can be calculated from the R matrix and c_j.

To produce one unit of the j^{th} product r_{ij}/r_{jj} units of the i^{th} product are needed, and the cumulative manufacturing costs have already been added to the i^{th} product. Thus, the sum, $\sum_{i=1}^{j-1} (r_{ij}/r_{jj})s_i$ is the total intermediate value of the inputs required for producing one unit of the j^{th} product. Additionally the cost, c_j, is added to one unit of the j^{th} product while processing at the j^{th} operation. For each of the m raw material suppliers, s_j equals c_j because there is no input to each supplier.

Recursive expressions for s_j are:

(3-1) $$s_j = c_j \quad \text{for } 1 \leqslant j \leqslant m$$

(3-1a) $$s_j = \sum_{i=1}^{j-1} (r_{ij}/r_{jj})s_i + c_j \quad \text{for } m + 1 \leqslant j \leqslant n$$

<u>Probabilistic Behaviors of Product</u>

The quality of intermediate product depends on two factors. One is the operational error, e_j, at each operation and the other is the decision made at each potential inspection station.

If inspection is performed at an inspection station, the quality of intermediate product from the j^{th} operation will be affected by the inspector's errors, and if not, it remains unchanged until later operations are involved.

Figure 3-3 is a generalized diagram of material flows and related errors at the j^{th} operation and inspection station.

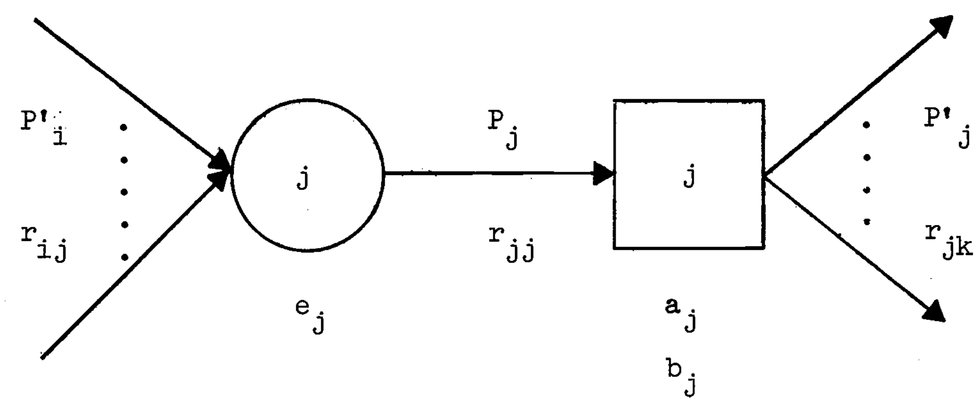

Figure 3-3. Material Flows at the j^{th} Operation and Inspection Station

The inputs with different flow rates as well as different quality levels flow into the j^{th} operation where the operational error, e_j, is assumed. Each event that each unit of the j^{th} product attains a type-j defect is assumed to be mutually independent. Besides, it is also independent of previous defects attained.

Let P_j = Pr (A unit of product after the j^{th} operation has at least one defect),

P'_j = Pr (A unit of product after the j^{th} inspection station has at least one defect),

O_j = The event that a unit of the j^{th} product is defect-free immediately after the j^{th} operation, and

I_j = The event that a unit of the j^{th} product is defect-free immediately after the j^{th} inspection station.

Then,

(3-2) $$P_j = 1 - \Pr\ (O_j) \quad \text{for } j = 1,2\ldots,n, \text{ and}$$

(3-3) $$P'_j = 1 - \Pr\ (I_j) \quad \text{for } j = 1,2,\ldots,n.$$

As discussed earlier r_{ij}/r_{jj} units of the i^{th} product are needed to produce one unit of the j^{th} product, and therefore, the condition that all r_{ij}/r_{jj} units of the i^{th} product are defect-free is necessary for the j^{th} product to be good. In addition, there should be no operational errors to produce a defect-free product at the j^{th} operation.

Thus,

Pr (O_j) = Pr (no operational errors occur at the j^{th} operation)

(3-4) x Pr (r_{ij}/r_{jj} units of the i^{th} product are defect-free for all $i < j$) for $j = m+1,\ldots,n$.

Because each event that each unit of inputs is defect-free is mutually independent, and $\Pr\ (A \cap B) = \Pr\ (A) \times \Pr\ (B)$ for two mutually independent events A and B,

Pr (r_{ij}/r_{jj} units of the i^{th} product are defect-free for all $i<j$)

(3-4a) $$= \prod_{i<j} \Pr\ (I_i)^{r_{ij}/r_{jj}}$$

$$= \prod_{i<j} (1 - P'_i)^{r_{ij}/r_{jj}} \quad \text{from Eq. (3-3).}$$

Besides,

Pr (No operational errors occur at the j^{th} operation)

(3-4b) $\quad = (1 - e_j)$ for $j = 1,2,\ldots,n$

From Eq. (3-4), (3-4a), and (3-4b),

(3-4c) $$\Pr(O_j) = (1-e_j)\prod_{i<j}(1-P'_i)^{r_{ij}/r_{jj}}$$

for $j = m+1,\ldots,n$.

Therefore, from Eq. (3-2) and (3-4c)

(3-5) $$P_j = 1 - (1 - e_j)\prod_{i<j}(1 - P'_i)^{r_{ij}/r_{jj}}$$

for $j = m+1,\ldots,n$.

If $1 \leqslant j \leqslant m$, Eq. (3-5) is reduced to Eq. (3-5a) because there is no input to the j^{th} raw material supplier.

(3-5a) $\quad P_j = e_j \quad$ for $1 \leqslant j \leqslant m$

Eq. (3-5) and (3-5a) describe the probabilistic behaviors of product immediately after the j^{th} operation for $j = 1,2,\ldots,n$. However, they cannot completely describe the probabilistic behavior of product progressing through the process. To be complete the probabilistic behaviors of product immediately after an inspection station should be described according to the following procedures.

All the j^{th} product are sent to the j^{th} insepction station where all-or-none screening inspection is applied with Type I and Type II inspector's errors. A Type I error occurs when a good item is classified as a defective item while a Type II error occurs when a defective item is classified as a good item by an inspector.

Let a_j = Pr (Type II errors occur at the j^{th} inspection station),

b_j = Pr (Type I errors occur at the j^{th} inspection station),

G_j = the event that one unit of the j^{th} product is defect-free immediately after the j^{th} operation,

B_j = the event that one unit of the j^{th} product is defective immediately after the j^{th} operation,

A_j = the event that one unit of the j^{th} product is accepted by the j^{th} inspector, and

E_j = the event that one unit of the j^{th} product is rejected by the j^{th} inspector.

Figure 3-4 shows the relationships between events mentioned above, and the related probabilities can be expressed as follows:

$$\Pr(B_j) = 1 - \Pr(G_j) = P_j,$$

$$\Pr(A_j/B_j) = a_j, \quad \text{and}$$

$$\Pr(E_j/G_j) = b_j.$$

If inspection is performed at the j^{th} inspection station, P'_j can be expressed as follows:

$$(3\text{-}6) \quad P'_j = \frac{\text{Expected number of defectives remaining immediately after the } j^{th} \text{ inspection station}}{\text{Total number of the } j^{th} \text{ product immediately after the } j^{th} \text{ inspection station.}}$$

The denominator of Eq. (3-6) is the same as the total number of the j^{th} product delivered to the j^{th} inspection station because it is assumed that detected defective items are replaced by good ones immediately. Besides, the numerator of Eq. (3-6) is the same as the expected number of defectives accepted by the j^{th} inspector. Therefore, Eq. (3-6) is equivalent to:

$$(3\text{-}6a) \quad P'_j = \frac{\left[r_{ij} \Pr(B_j)\right] \Pr(A_j/B_j)}{r_{jj}}$$

$$= \Pr(B_j) \; \Pr(A_j/B_j)$$

$$= P_j \, a_j \qquad \text{for } j = 1,2,\ldots,n.$$

Another way of determining P'_j is to directly utilize Figure 3-4. After performing inspection and replacing detected defective items by good ones, the portion of defective items in the total number of items delivered to inspection is the same as $\Pr(A_j \cap B_j)$ as shown in Figure 3-5. Applying the Bayes' Rule,

$$\Pr(A_j \cap B_j) = \Pr(B_j) \; \Pr(A_j/B_j) = P_j \, a_j$$

which is the same as Eq. (3-6a).

	B_j	G_j
A_j	$A_j \cap B_j$	$A_j \cap G_j$
E_j	$E_j \cap B_j$	$E_j \cap E_j$

Figure 3-4. Relationships Between Events at the j^{th} Inspection Station

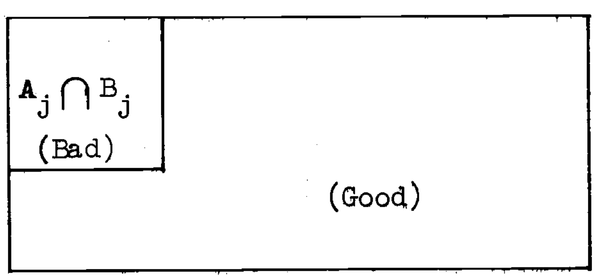

Figure 3-5. Portions of Bad and Good Items After the j^{th} Inspection

Note that P'_j does not depend upon Type I errors as shown in Eq. (3-6a). This is true because good items scrapped due to Type I errors are eventually replaced by good ones, and therefore, Type I errors have no influence on the number of defective items that remain.

If inspection is not performed at the j^{th} inspection station,

(3-7) $P'_j = P_j$ for $j = 1,2,\ldots,n.$

Eq. (3-6a) and (3-7) can be expressed at the same time with respect to the j^{th} decision variable, d_j or d'_j. As mentioned earlier, if inspection is performed at the j^{th} inspection station, $d_j = 1$, and if not, $d_j = 0$. Besides, $d'_j = 1 - d_j$ for $j = 1,2,\ldots,n$. Therefore, Eq. (3-6a) and (3-7) are combined into:

(3-8) $$P'_j = P_j \left[1 - (1 - a_j)\, d_j\right]$$
$$= P_j \left[a_j + (1 - a_j)\, d'_j\right] \quad \text{for } j = 1,2,\ldots,n.$$

Thus, the probabilistic behaviors of products progressing through a production process are completely described by Eq. (3-5), (3-5a), and (3-8). Above all things, P'_n is especially important because it represents the average outgoing quality of final product which is required not to be greater than the predetermined quality level, P_{AOQ}.

Quality Costs Structures

From the assumptions made about the nature of quality costs in General section, it is straightforward to express each of them as a function of decision variables.

Let l_j = Sum of appraisal costs per unit of the j^{th} product, and

v_j = Salvage value per unit of the j^{th} product.

It is assumed that $v_j < s_j$ for all j, and therefore, $s_j - v_j$ is the actual cost incurred by scrapping one unit of the j^{th} product.

Then, each group of quality costs needed to produce one unit of final product can be expressed as follows:

$$TAC = \text{Total Appraisal Costs} = \sum_{j=1}^{n} r_{jj}\, l_j\, d_j$$

$$(3\text{-}9) \qquad = \sum_{j=1}^{n} r_{jj}\, l_j\, (1-d'_j)$$

TIC = Total Internal failure Costs

$$= \sum_{j=1}^{n} r_{jj}\, Pr\,(E_j)\,(s_j - v_j)\, d_j$$

$$(3\text{-}10) \qquad = \sum_{j=1}^{n} r_{jj}\, Pr\,(E_j)\,(s_j - v_j)\,(1 - d'_j)$$

where
$$Pr\,(E_j) = Pr\,(E_j \cap B_j) + Pr\,(E_j \cap G_j)$$
$$= Pr\,(B_j)\; Pr\,(E_j/B_j) + Pr\,(G_j)\; Pr\,(E_j/G_j)$$
$$= P_j\,(1 - a_j) + (1 - P_j)\, b_j$$
$$= b_j + P_j\,(1 - a_j - b_j).$$

TEC = Total External failure Costs

= (Expected external failure costs per defective unit)

(3-11) x Pr (A unit of final product is defective)

$$= k\, P'_n$$

Therefore,

Total expected quality costs per unit of final product

=TAC + TIC + TEC.

Mathematical Model

In the General section the problem is defined as to find an optimal policy vector d or d' which minimizes the total quality costs while achieving the required average outgoing quality level of final product. Based upon the definition of the problem and the results obtained in the previous sections, a mathematical model is formulated. In the model, the objective function represents the total quality costs per unit of final product, and the constraint represents the requirement of AOQL of the final product. For convenience, the policy vector d' is adopted instead of d. Therefore, the problem is to:

Minimize Z (d')

$$= \sum_{j=1}^{n} r_{jj}\, l_j\, (1-d'_j) + \sum_{j=1}^{n} r_{jj} \left[b_j + P_j\, (1 - a_j - b_j)\right] (s_j-v_j)$$

$$(1 - d'_j) + kP'_n$$

subject to

$$P'_n \leqslant P_{AOQ}$$

$$d'_j = 0 \text{ or } 1 \qquad \text{for } 1 \leqslant j \leqslant n.$$

where

$$(3\text{-}5) \quad P_j = 1-(1-e_j) \prod_{i<j} (1-P'_i)^{r_{ij}/r_{jj}} \quad \text{for } m+1 \leqslant j \leqslant n$$

$$(3\text{-}5a) \quad = e_j \quad \text{for } j \leqslant m$$

$$(3\text{-}8) \quad P'_j = P_j \left[a_j + (1 - a_j)\, d'_j\right] \qquad \text{for } 1 \leqslant j \leqslant n$$

The cumulative manufacturing costs, s_j, are the same as defined in Eq. (3-1) and (3-1a). Definitions of all constants and variables are included in Appendix A.

It is clear that the problem is formulated as a nonlinear zero-one integer programming problem because each P_j or P'_j in the objective function as well as in the constraint is itself a nonlinear function of the decision variables. To appreciate the nature of the problem further, the objective function is rewritten as the difference of two functions, and an important theorem related with P_j and P'_j is established as follows:

The revised objective function:

Minimize Z (d')

$$= Z_1 (d') - Z_2 (d')$$

$$= \sum_{j=1}^{n} \left[r_{jj}\, 1_j + r_{jj} \quad (s_j - v_j) \left\{ b_j + P_j + P_j (a_j + b_j)\ d'_j \right\} \right] + kP'_n$$

$$- \sum_{j=1}^{n} \left[r_{jj} 1_j d'_j \ + r_{jj}\ (s_j - v_j) \ \left\{ b_j d'_j + P_j (a_j + b_j) + P_j d'_j \right\} \right]$$

<u>Theorem</u>. Each P_j or P'_j is monotone nondecreasing in each of the decision variables $d'_1, d'_2, \ldots\ldots, d'_n$.

To prove the theorem consider a policy vector d', and assume that any one of its components, say d'_k is increased from 0 to 1. Then,

<u>Lemma 1</u>. $P'_k/_{d_k=1} \geqslant P'_k/_{d_k=0}$ for $k = 1,2,\ldots\ldots,n$.

Proof. From Eq. (3-8),

(3-12a) $P'_k/_{d'_k=0} = P_k a_k = P_k$ because $0 \leqslant a_k \leqslant 1$, and

(3-12b) $P'_k/_{d'_k=1} = P_k$ for $k = 1,2,\ldots\ldots,n$.

From Eq. (3-12a) and (3-12b), the Lemma is proved.

<u>Lemma 2</u>. $P_j/d'_k{=}1 = P_j/d'_k{=}0$ for $j \leqslant k$.

Proof. This is intuitively clear because, if $j \leqslant k$, the outputs from the j^{th} operation are delivered to the k^{th} inspection station directly or indirectly, and therefore, P_j, which has been already determined by the decisions previously made, does not depend upon d'_k.

<u>Lemma 3.</u> $P'_j/d'_k{=}1 = P'_j/d'_k{=}0$ for $j < k$.

Proof. This is also clear based upon the similar reason in Lemma 2.

<u>Lemma 4</u>. $P_j/d'_k{=}1 \geqslant P_j/d'_k{=}0$ for $j > k$.

Proof. If $k < j \leqslant m$, from Eq. (3-5a),

$$(3\text{-}13) \quad P_j/d'_k{=}1 = P_j/d'_k{=}0 = e_j \text{ for } k<j \leqslant m$$

because there is no interaction between raw material suppliers.

If $j>k$ and $j>m$, from Eq. (3-5),

$$(3\text{-}14a) \quad P_j/d'_k{=}1 = 1 - (1-e_j)(1 - P'_k/d'_k{=}1)^{r_{kj}/r_{ij}} \left[\prod_{\substack{i<j \\ i \neq k}} (1 - P'_i)^{r_{ij}/r_{jj}}\right]$$

$$(3\text{-}14b) \quad P_j/d'_k{=}0 = 1 - (1-e_j)(1 - P'_k/d'_k{=}0)^{r_{kj}/r_{jj}} \left[\prod_{\substack{i<j \\ i \neq k}} (1 - P'_i)^{r_{ij}/r_{jj}}\right]$$

From Lemma 1,

$$(3\text{-}15a) \quad 1 - P'_k/d'_k{=}1 \leqslant 1 - P'_k/d'_k{=}0 \quad \text{and}$$

$$(3\text{-}15b) \quad (1 - P'_k/d'_k{=}1)^{r_{kj}/r_{jj}} \leqslant (1 - P'_k/d'_k{=}0)^{r_{kj}/r_{jj}}$$

because r_{kj}/r_{jj} is either some positive integer or 0 according to whether there is a path between the k^{th} inspection station and the j^{th} operation or not. Applying the inequality (3-15b) to Eq. (3-14a) and

(3-14b) yields:

(3-16) $\quad P_j/_{d'_k=1} \geqslant P_j/_{d'_k=0}$ for $j>k$ and $j>m$

because the other corresponding terms in Eq. (3-14a) and (3-14b) are the same. From Eq. (3-13) and from the inequality (3-16), the Lemma is proved.

<u>Lemma 5</u>. $P'_j/_{d'_k=1} \geqslant P'_j/_{d'_k=0}$ for $j > k$.

Proof. From Eq. (3-8),

(3-17a) $\quad P'_j/_{d'_k=1} = P_j/_{d'_k=1} \left[a_j + (1-a_j)\, d'_j\right]$ and

(3-17b) $\quad P'_j/_{d'_k=0} = P_j/_{d'_k=0} \left[a_j + (1-a_j)d'_j\right]$.

From Lemma 4, $P_j/_{d'_k=1} \geqslant P_j/_{d'_k=0}$. The Lemma is proved.

From Lemma 2 and Lemma 4, P_j for $j=1,2,\ldots,n$ is monotone nondecreasing in each of the decision variables d'_1, $d'_2\ldots,d'_n$. P'_j for $j=1,2,\ldots,n$ is also monotone nondecreasing in each of decision variables by Lemma 1, Lemma 3, and Lemma 5. Thus, the theorem is proved.

Using the theorem the following important characteristics of the mathematical model are derived.

1. The left hand side of the constraint, P'_n, is monotone nondecreasing in each of decision variables $d'_1, d'_2, \ldots, d'_n$.

2. $Z_1(d')$ and $Z_2(d')$ in the revised objective function are monotone nondecreasing in each of decision variables $d'_1, d'_2, \ldots d'_n$ because multiplications and additions of nondecreasing functions which have nonnegative values are also nondecreasing. Note that each decision variable d'_j is also nondecreasing in each of decision variables. Therefore, the objective

function of the mathematical model can be expressed as the difference of two monotone nondecreasing functions.

The above two characteristics of the model are essential for providing efficient solution techniques for the problem.

Further Remarks on the Model

The model can be easily expanded to the following cases with slight modifications.

1. The case in which several different final operations are allowed.
2. The case in which the availability of inspection effort is limited.
3. The case in which inspection cannot be performed at certain inspection stations due to physical or technical difficulties.

Consider a nonserial production system with n-q+1 final operations where n-q+1 different types of final product are produced as shown in Figure 3-6. It is assumed that the desired quality of each final product is $(P_{AOQ})_j$ for j=q, q+1,...,n. In addition, the expected external failure costs for each unit of final product is assumed to be k_j for j=q, q+1,...,n. Then, the problem can be modified as follows:

Minimize Z(d')

$$=\sum_{j=1}^{n} r_{jj}\, l_j(1-d'_j)+\sum_{j=1}^{n} r_{jj}\left[b_j+P_j(1-a_j-b_j)\right] + (s_j-v_j)(1-d'_j)$$

$$+\sum_{j=n-q+1}^{n} k_j\, P'_j$$

subject to

$$P'_j \leqslant (P_{AOQ})_j \qquad \text{for } j = q,\ q+1,\ldots,n.$$

$$d'_j = 0 \text{ or } 1 \qquad \text{for } 1 \leqslant j \leqslant n$$

Note that the objective function as well as constraints in the new problem is still monotone nondecreasing in each of the decision variables d'_1, $d'_2,\ldots,$ d'_n.

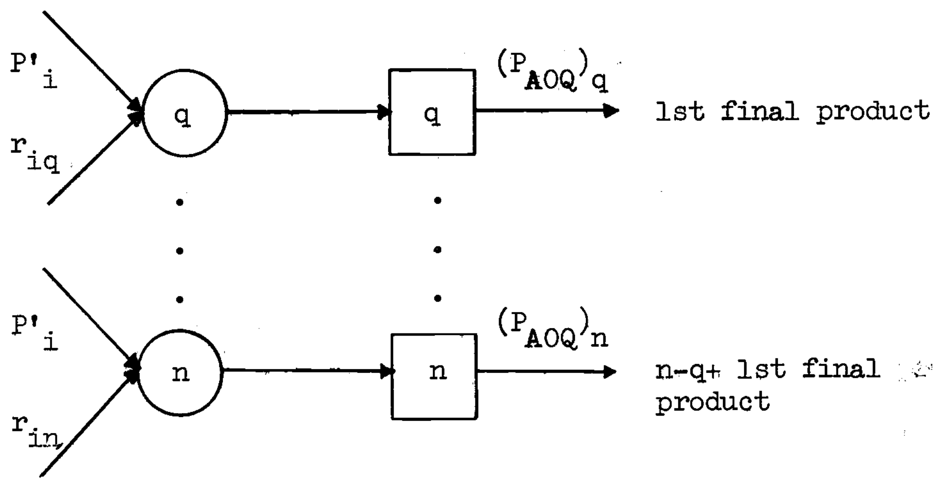

Figure 3-6. A Nonserial Production System with Multiple Final Operations

Suppose that the number of available qualified inspectors is limited to w. Then, another new problem can be formulated by adding a new constraint:

$$(3\text{-}18) \quad \sum_{j=1}^{n} d_j = \sum_{j=1}^{n} (1-d'_j) = n - \sum_{j=1}^{n} d'_j \leqslant w$$

The above inequality can be rewritten as:

(3-18a) $$-\sum_{j=1}^{n} d'_j = w-n$$

It is clear that the sum $\sum_{j=1}^{n} d'_j$ is monotone nondecreasing in each of decision variables.

If it is impossible to perform inspection at the j^{th} inspection station, this limitation can be formulated in the model by adding the following constraint:

(3-19) $$d_j = (1-d'_j) = 0$$

Eq. (3-19) can be expressed by two inequalities as usual, i.e.,

(3-19a) $$d'_j \leqslant 1$$

(3-19b) $$-d'_j \leqslant -1$$

Obviously the term d'_j in the right hand side of the above two inequalities (3-19a) and (3-19b) is monotone nondecreasing in each of decision variables.

Based upon the above discussions, it is clear that the model is flexible enough to include various types of limitations and requirements existing in general production-inspection systems. Moreoever, the important characteristics that the objective function and the constraints are monotone nondecreasing in each of decision variables can be reversed. Based upon these characteristics of the problem, two solution techniques are proposed in Chapter IV, and the problem is analyzed in Chapter V using the original model and the chosen solution technique.

IV. SOLUTION METHODS

General

Although there have been extensive theoretical contributions regarding the solution of integer programming problems, progress in the computational aspects has not been remarkable in spite of the rapid evolution of digital computers. Besides, each current algorithm is usually designed to solve a specific type of problem, and this deficiency becomes severer when nonlinearity is encountered. Considering the difficulties currently existing, it seems most desirable to solve nonlinear integer programming problems according to the following steps:

1. Analyze the nature of the problem to be solved.
2. Determine the most adequate approach for the problem.
3. Choose a specific algorithm which is most suitable to the problem.
4. Modify the algorithm if desired to make it more suitable to the problem.

As discussed in Chapter III, the formulated problem has unique characteristics. In brief, the objective function as well as the constraint is nonlinear and monotone nondecreasing in each of the decision variables which takes on values of either 0 or 1. Besides, it is clear that the objective function and the constraint are some polynomials of decision variables, however, it is quite difficult to express them explicitly in terms of these decision variables. Thus, such analytical approaches as linear transformation of the problem, solving the problem directly by relaxing integer constraints, etc. do not seem promising.

Methods for solving integer programming problems are generally categorized into two groups: search methods and cutting plane methods. Cutting plane methods are not adequate for the formulated problem because they are developed primarily for linear integer programming problems. The idea of search methods is to implicitly enumerate all possible solution points which are finite. The difference between search methods and exhaustive enumeration is that search methods develop techniques to enumerate only a small portion of all possible solutions while automatically excluding the remaining solution points as unpromising.

Search methods include implicit enumeration and the branch-and-bound technique. Implicit enumeration is especially efficient for zero-one problems. The branch-and-bound technique solves the problem by relaxing the integer constraints and by setting up upper and lower bound on the objective function at each iteration. If the problem to be solved is a linear program the branch-and-bound procedure may be efficient, however, the formulated problem is a nonlinear program which is usually hard to solve, and worse, it is difficult to express the objective function as well as the constraint explicitly in terms of decision variables. However, for zero-one implicit enumeration an explicit expression of the objective function in terms of the decision variables is not needed, and therefore, it is chosen as the most adequate approach for solving this formulated problem.

Based upon the above discussion, two specific algorithms have been proposed. One is Lawler and Bell's zero-one enumeration scheme (Lawler and Bell, 1967), and the other is Glover's partial enumeration scheme (Glover, 1965). In addition to the fact that the requirements of the two algorithms are satisfied by the formulated problem, they have several advantages. They do not require explicit expressions of the objective function as well as constraints with respect to decision variables. Moreover, they always give true optimal solutions, and can be more easily programmed than any other algorithms.

Glover's scheme is modified to make it suitable to the problem, and the computational efficiency of the two algorithms is evaluated in Chapter V.

Lawler and Bell's Algorithm

Lawler and Bell's algorithm can be applied to any problem of the form:

Minimize $Z(d') = Z_1(d') - Z_2(d')$

subject to

$$g_{i1}(d') - g_{i2}(d') \leq b_i \text{ for } i=1,2,\ldots,m.$$

where

$$d'=(d'_n, d'_{n-1}, \ldots, d'_1)$$

$$d'_j = 0 \text{ or } 1, \; j = 1,2,\ldots,n.$$

A restriction of the algorithm on the above problem is that each of the functions Z_1, Z_2, g_{i1}'s, and g_{i2}'s is monotone nondecreasing in each of binary variables $d'_1, d'_2, \ldots, d'_n$. Our problem exactly satisfies this condition as discussed in Chapter III. In our problem, there is only one constraint in which $g_{11}(d') = P'_n$, $g_{12}(d') = 0$, and $b_1 = P_{AOQ}$.

In order to describe the algorithm, some preliminary explanation is required. Consider binary vectors x and y. It is said that $x \leqslant y$ if and only if $x_j \leqslant y_j$ for all j. A binary vector d' is said to have the numerical order:

$$(4\text{-}1) \qquad N(d') = 2^0 d'_n + 2^1 d'_{n-1} + \dots + 2^{n-1} d'_1.$$

Suppose that n=5, and then, all possible solution points (2^5 = 32) can be arranged in numerical order, i.e.,

Solution	N
(0,0,0,0,0)	1
(0,0,0,0,1)	2
(0,0,0,1,0)	3
(0,0,0,1,1)	4
(0,0,1,0,0)	5
(0,0,1,0,1)	6
(0,0,1,1,0)	7
(0,0,1,1,1)	8
(0,1,0,0,0)	9
⋮	⋮
(1,1,1,1,1)	31

As shown in the above example, immediately following a certain binary vector d', there may or may not exist binary vector(s) d such that $d' \leqslant d$. Let d'* be the first vector following d' in the numerical ordering such that $d' \nleqslant d'^*$. For example, for a vector d'=(0,0,1,0,0),

$d=(0,0,1,0,1)$, $(0,0,1,1,0)$, $(0,0,1,1,1)$ are greater than d', and $d'^* = (0,1,0,0,0)$ is the first vector such that $d' \nleqslant d'^*$. The vector d'^* for a vector d' can be generated as follows:

1) Generate $d'-1$. The numerical order of $d'-1$ is less than that of d' by 1.
2) Logically "or" d' and $d'-1$ to obtain d'^*-1.
3) Add 1 to d'^*-1 to obtain d'^*.

For example, $d'=(0,0,1,0,0)$, $d'-1=(0,0,0,1,1)$, $d'^*-1=(0,0,1,1,1)$, and $d'^*=(0,1,0,0,0)$.

The algorithm implicitly enumerates 2^n possible solutions starting with $d'=(0,0,\ldots,0)$ and terminating when $d'=(1,1,\ldots,1)$. In the process of enumeration, proper fathoming tests are provided to discard unpromising solutions as much as possible. These fathoming tests can be described as follows.

Let d' be the vector to be considered, $\bar{Z}$ be the minimum objective value so far attained. Then,

<u>Test 1</u>: If $g_{i1}(d')-g_{i2}(d'^*-1) > b_i$ for any i,

skip to d'^*.

<u>Test 2</u>: If $Z_1(d')-Z_2(d'^*-1) > \bar{Z}$, skip to d'^*.

The idea of the above fathoming tests is based upon the monotone nondecreasing nature of the objective function and constraints. Test 1 is justified because $g_{i1}(d')-g_{i2}(d'^*-1)$ is the minimum attainable value of $g_{i1}(d)-g_{i2}(d)$ for any d in the interval $[d', d'^*-1]$. To prove this, consider the following inequalities.

$$(4\text{-}2) \quad g_{i1}(d') \leqslant g_{i1}(d) \text{ for any } d \text{ in } [d', d'^*-1].$$

Inequality (4-2) is true because g_{i1} is monotone nondecreasing and $d' \leqslant d$ for any d in the interval.

$$(4\text{-}3) \quad g_{i2}(d'^*-1) \geqslant g_{i2}(d) \quad \text{for any } d \text{ in } [d', d'^*-1] .$$

Inequality (4-3) is also true because g_{i2} is monotone nondecreasing and $d'^*-1 \geqslant d$ for any d in the interval. Multiplying each side of inequality (4-3) by -1 yields:

$$(4\text{-}3a) \quad -g_{i2}(d'^*-1) \leqslant -g_{i2}(d) \text{ for any } d \text{ in } [d', d'^*-1] .$$

Adding inequality (4-2) to (4-3a) yields:

$$(4\text{-}4) \quad g_{i1}(d') - g_{i2}(d'^*-1) \leqslant g_{i1}(d) - g_{i2}(d) \quad \text{for any } d \text{ in } [d', d'^*-1] .$$

Thus, if $g_{i1}(d') - g_{i2}(d'^*-1) > b_i$ for any i, there is no vector d in the interval such that $g_{i1}(d) - g_{i2}(d) \leqslant b_i$ for all i, and therefore, the interval $[d', d'^*-1]$ can be discarded without further consideration. Test 2 can also be justified through the same procedures.

In our problem, there is only one constraint as mentioned earlier, and therefore, the Test 1 can be reduced to:

<u>Test 1</u>: If $P'_n(d') > P_{AOQ}$, skip to d'^*.

If a vector d' passes the two fathoming tests, the enumeration continues by increasing the numerical order of d' by 1, because the interval $[d', d'^*-1]$ deserves further consideration.

Aside from the above fathoming tests, there should be a stopping rule to terminate the algorithm. The rule is:

<u>Stopping Rule</u>: If $d'=(1,1,\ldots,1)$, stop.

Besides, the run can also be terminated if the initial vector $d'=(0,0,\ldots,0)$ does not pass Test 1. This is true because any other vectors to be considered are greater than the initial vector, and therefore, they cannot pass Test 1 either.

The flowchart and computer program for Lawler and Bell's algorithm are illustrated in Appendix B.

As an example, the following problem is solved by the algorithm.

Minimize $Z(d)=Z_1(d)-Z_2(d)$

$$=(d_1^2+2d_2^2+3d_5)-(d_3d_4+d_2d_5-d_4)$$

Subject to

$$4d_3^2+3d_1+2d_2+d_3d_4d_5 \leqslant 2$$

$$d_j=0 \text{ or } 1 \text{ for } j=1,2,\ldots,5.$$

Solution procedures follow the steps described in Appendix B.2 as shown in Table 4-1. In this table, $\bar{d}$ represents an optimal solution so far attained. After nine iterations an optimal solution $d=(0,1,0,0,0)$ is obtained, and the optimal value of the objective function is -1.

Modified Glover's Enumeration Scheme

Glover's zero-one enumeration scheme is general because it does not depend upon the form of the zero-one problem. Any linear or nonlinear zero-one integer program can be solved by this scheme as long as proper fathoming tests can be provided.

In order to describe Glover's algorithm, the following terminologies are explained.

1. Complete Solution: A solution in which all variables are fixed at binary values.

Table 4-1. An Example of Solution Procedures of Lawler and Bell's Algorithm

d (d_5,d_4,d_3,d_2,d_1)	d*-1	Procedures
(0,0,0,0,0)	(0,0,0,0,0)	feasible; set $\bar{d}$=(0,0,0,0,0); set $\bar{Z}$=0; set d'=(0,0,0,0,1).
(0,0,0,0,1)	(0,0,0,0,1)	infeasible; skip to d*.
(0,0,0,1,0)	(0,0,0,1,1)	feasible; not optimal; skip to d*.
(0,0,1,0,0)	(0,0,1,1,1)	infeasible; skip to d*.
(0,1,0,0,0)	(0,1,1,1,1)	feasible; optimal; set $\bar{Z}$=-1; set $\bar{d}$=(0,1,0,0,0); d=d+1.
(0,1,0,0,1)	(0,1,0,0,1)	infeasible; skip to d*.
(0,1,0,1,0)	(0,1,0,1,1)	feasible; not optimal; skip to d*.
(0,1,1,0,0)	(0,1,1,1,1)	infeasible; skip to d*.
(1,0,0,0,0)	(1,1,1,1,1)	feasible; not optimal; skip to d*. Because d*-1=(1,1,1,1,1), stop.

2. Partial Solution: A set of variables with fixed binary values. Suppose that $d=(d_5, d_4,\ldots,d_1)$. If d_5, d_4, and d_3 are fixed at 0, 1, and 1, respectively, then, $(d_5, d_4, d_3) = (0,1,1)$ is a partial solution.

3. Free Variable: A variable which is not assigned a specific binary value. In the above example, d_2 and d_1 are free to take either 0 or 1.

4. Completion: A complete solution determined by assigning specific binary values to each free variable. The partial

solution (0,1,1) in the above example has four completions of (0,1,1,0,0), (0,1,1,0,1), (0,1,1,1,0) and (0,1,1,1,1).

5. Zero-Completion: A completion in which all free variables are fixed at 0. In the above example, (0,1,1,0,0) is the zero-completion of (0,1,1).

6. Unit-Completion: A completion in which all free variables are fixed at 1. In the above example, (0,1,1,1,1) is the unit-completion of (0,1,1).

The algorithm starts with an initial partial solution in which one variable is fixed at 0 or 1. Through proper fathoming tests provided a partial solution is evaluated whether it has promising completions or not. When all its completions can be discarded as nonpromising a partial solution is said to be fathomed, and a backward move is made to form a new partial solution. If it is believed that the partial solution has any promising completions, a forward move is made to search those completions. In the process of such augmentations, some solution points are discarded as nonpromising without explicit consideration, and the algorithm terminates when all complete solutions are enumerated explicitly or implicitly.

In order to avoid cumbersome bookkeeping procedures for scanning partial solutions, and to reduce the need for computer storages, Glover's scheme for generating the next partial solutions is modified by adopting systematic procedures for backward and forward moves. The modified algorithm starts with d'_n being fixed at 1. A forward move is made by fixing one of the free variables at 1. The free variable chosen always has a bigger subscript than any of the other remaining free variables. If a partial solution is proved to have no promising completions, a backward move is made by replacing the right-most element of a partial solution by 0.

An illustration of these systematic procedures for backward and forward moves is in Figure 4-1. To generate a next partial solution it is only necessary to keep the current partial solution and a signal variable which indicates the location of the right-most 1 in the current partial solution. Note that all possibel solution points are implicitly or explicitly enumerated through this systematic procedure in Figure 4-1.

Fathoming tests provided for solving our problem by Glover's scheme are also based upon the fact that the objective function as well as the constraint are nondecreasing in each of the decision variables. They include:

<u>Test 1</u>: If the current partial solution is a complete solution, the partial solution is fathomed.

Let $d'+$ and $d'-$ be the unit-completion and the zero-completion of a partial solution Q currently considered, respectively. Let $\overline{Z}$ be the minimum objective value so far attained. Then,

<u>Test 2</u>: If $P'_n(d'-) > P_{AOQ}$, make a backward move. If not, make a forward move to form a new partial solution.

<u>Test 3</u>: If $Z_1(d'-) - Z_2(d'+) > \overline{Z}$, make a backward move. If not make a forward move to form a new partial solution.

Test 1 is justified because there are no other completions to be considered for that partial solution. In Figure 4-1 partial solutions h, i, o, and p are complete solutions, and therefore, they are

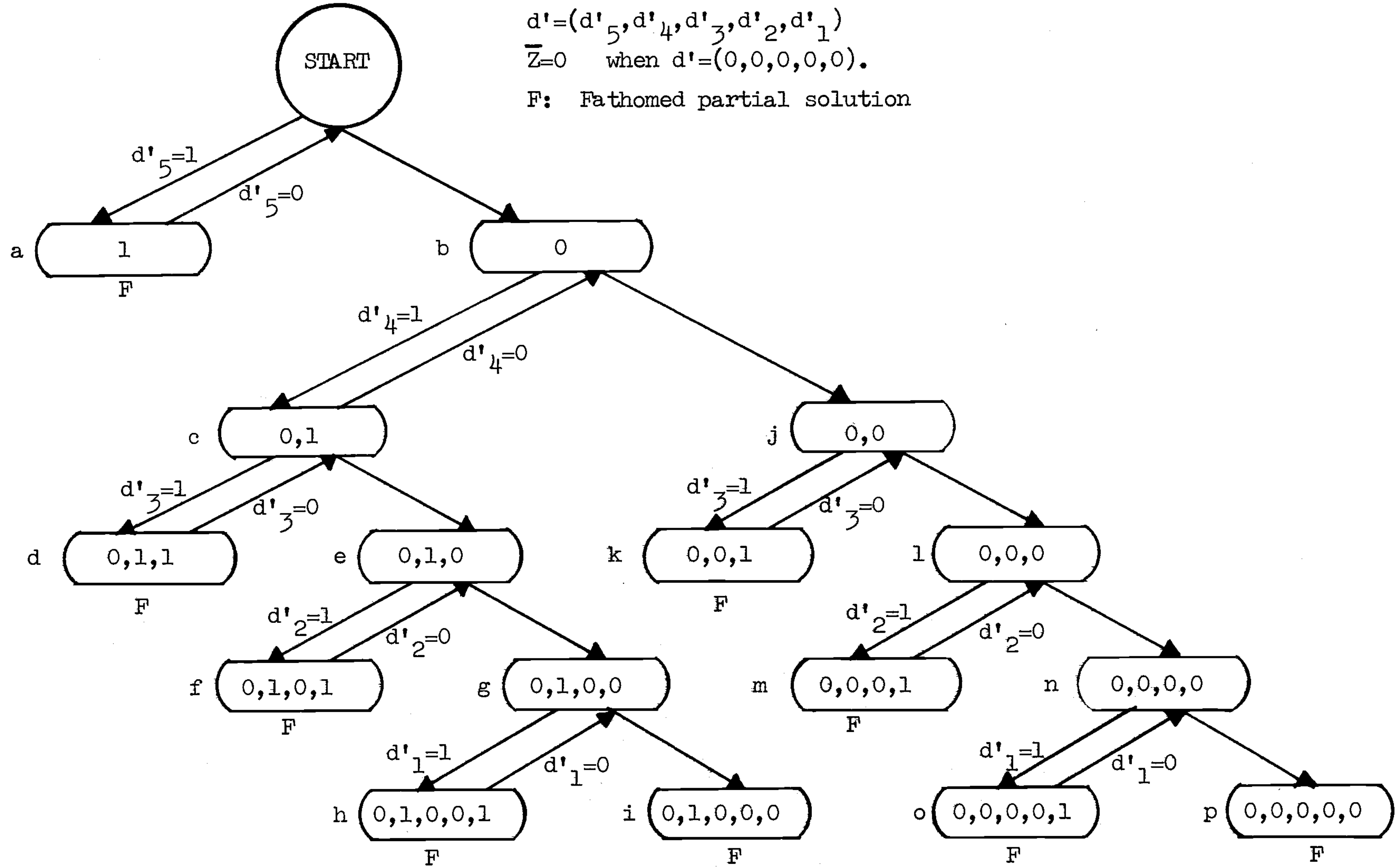

Figure 4-1. An Example of Forward and Backward Moves in Modified Glover's Enumeration Scheme.

fathomed. If the right-most element of each of these partial solutions is 1, then make a backward move to generate another partial solution which is complete. If the right-most element is 0, search the location of the right-most 1 in the partial solution. If there is no such element, the algorithm terminates. Let r be the location of the right-most 1, and then, replace the value of d'_r by 0 and delete all elements on the right of d'_r to generate a new partial solution. In Figure 4-1, partial solution j is generated from i according to the above procedures.

Test 2 and Test 3 can be justified because all completions of a partial solution Q lie in the interval $[d'-, d'+]$. Clearly, d'- is the smallest and d'+ is the biggest vector in the interval and $(d'-)^*-1=d'+$ according to the procedures for generating d'^*-1 in the previous section. Therefore, if $P'_n(d'-) > P_{APQ}$, no completions of the partial solution Q being considered can satisfy the feasibility because d'- is the smallest vector among all completions and P'_n is monotone nondecreasing in each of variables. If $Z_1(d'-) - Z_2(d'+) > \overline{Z}$, no completions can be optimal because $Z_1(d'-) - Z_2(d'+)$ is the minimum attainable value for any vector in the interval $[d'-, d'+]$. As an example, consider a partial solution c in Figure 4-1. For that partial solution d'-=(0,1,0,0,0) and d'+=(0,1,1,1,1). It passes the two fathoming tests, and forms a new partial solution d=(0,1,1) by a forward move. The partial solution d cannot pass either Test 2 or Test 3, and therefore, forms a new partial solution e=(0,1,0) by a backward move.

As mentioned earlier, the algorithm terminates when the current partial solution being considered is a complete solution in which all elements are fixed at zero. In Figure 4-1, the enumeration terminates after evaluating the partial solution p=(0,0,0,0,0).

Figure 4-2 is a simplified flowchart for modified Glover's scheme. A detailed flowchart and computer program are illustrated in Appendix C.

The major difference between the two algorithms is that vectors are enumerated lexicographically in Lawler and Bell's Algorithm while they are enumerated according to forward and backward moves in Glover's scheme. Besides, the procedure for generating (d'-)*-1 in Glover's scheme is much simpler than the procedure for generating d'*-1 in Lawler and Bell's algorithm because (d'-)*-1 can be generated by simply assigning 1's to all free variables in a partial solution.

The computational efficiency of the two algorithms is examined extensively in Chapter V.

Descriptions of Computer Programs

Two FORTRAN programs have been developed for the proposed two different algorithms. Flowcharts, listings of programs, and other detailed information about the programs are included in Appendix B and C.

Program LAWLER starts by reading in necessary data, and calculates the cumulative manufacturing costs at each operation as well as other necessary quantities. Initially, each element of the solution vector is set to zero, and a feasibility test is performed for that null vector. If it is infeasible, the run terminates by writing

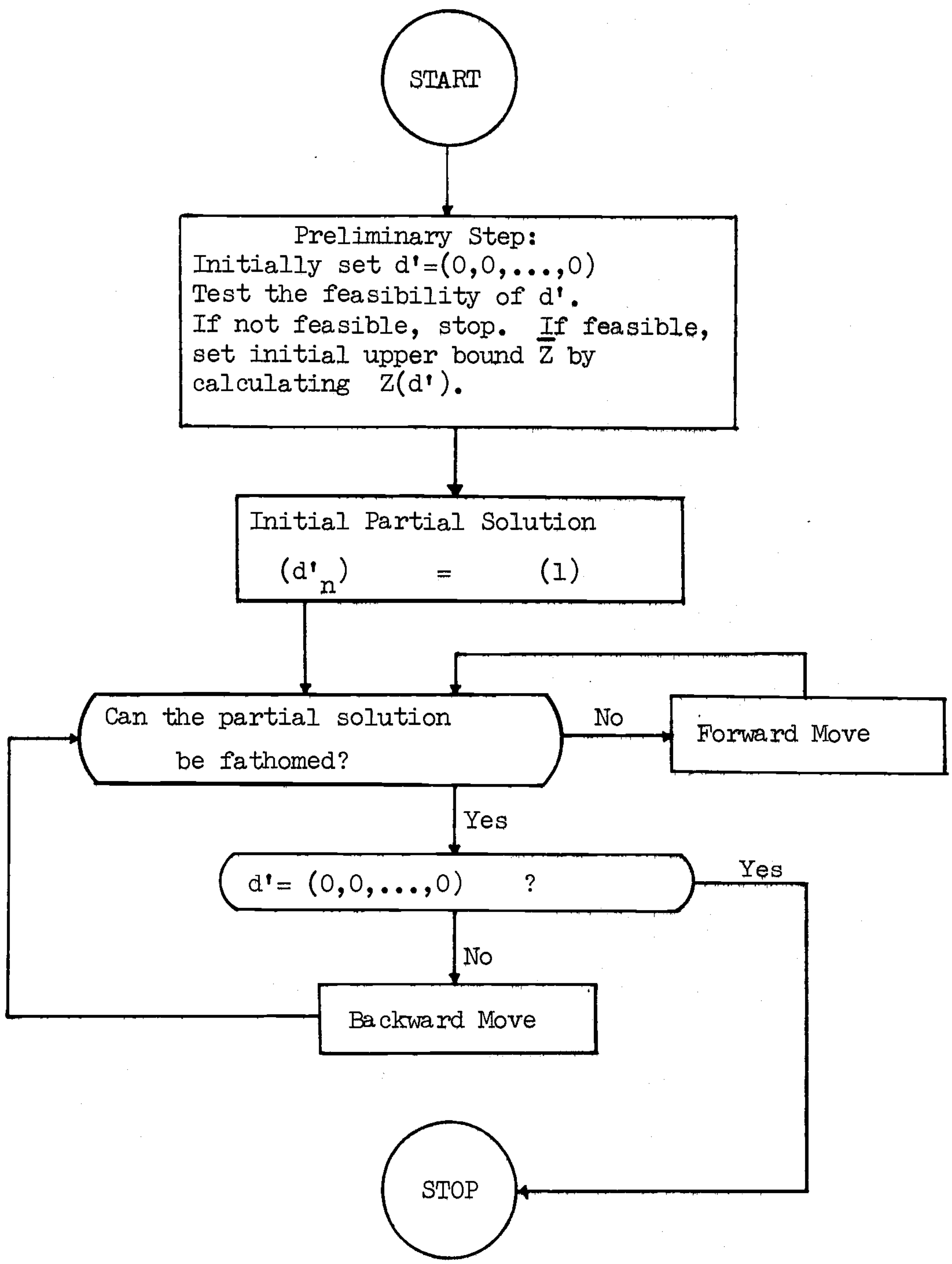

Figure 4-2. A Simplified Flow Chart of Modified Glover's Enumeration Scheme.

"INFEASIBLE" on the output. If it is feasible the null vector is saved in the array IDMI(I) by calling the subroutine EQUAL (ID, IDMI), and the objective value is also saved by the variable ZMIN as an initial upper bound. The next solution vector to be considered is generated by calling the subroutine DPLUS (ID), and the run continues according to the procedures described in Appendix B.2 until the final vector (1,1,...,1) is encountered.

The Fathoming Test 1 is performed by calling the subroutine FEASI(ID) and comparing PI(NP) to PAOQ. If a solution vector passes Test 1, the Fathoming Test 2 is performed by calling the subroutine ZTEST(ID, IDPR,Z) and comparing Z to ZMIN. If a solution vector passes those two tests the objective value for ID(I) is calculated by calling the subroutine OBJCT (ID,ZD), and ZD is compared to ZMIN. If ZD is less than ZMIN, ZMIN is replaced by ZD, and ID(I) is saved in the array IDMI(I) by calling the subroutine EQUAL(ID, IDMI). If ZD is greater than ZMIN another test is performed by calling OVFLW(ID,M1) to know whether the current solution vector reaches to the final vector or not. If it reaches to the final vector M1 becomes 1 and the run terminates. If not, the run continues by calling the subroutine DPLUS (ID) to update the current solution vector. If the current solution vector cannot pass either of those two fathoming tests, another test is performed by calling the subroutine OVFLW (IDPR,M2) to know whether the vector IDPR (I) reaches to the final vector or not. If IDPR(I) reaches to the final vector the run terminates, and if not, the run continues by calling the subroutine

DPLUS(IDPR) to generate the next solution vector to be considered. Throughout the run, the total number of vectors explicitly enumerated are counted by NOEN. The final content of IDMI(I) is converted so that the relationship $d'_j=1-d_j$ may be satisfied.

The initial stage of the program GLOVER is exactly the same as that of the program LAWLER, however, the null vector is updated to (1,0,...,0) rather than (0,0,...,1). In the program GLOVER, the zero-completion of a partial solution is considered instead of directly considering the partial solution itself. For example, a solution vector (0,1,1,0,0,0) in the program is identical to a partial solution (0,1,1).

In the program GLOVER, solution vectors are updated by forward or backward moves. For the convenience of those procedures, two variables, MR and IST are provided. MR indicates the location of the right-most 1 in the current solution vector, and IST indicates the location of the second right-most 1 in the current solution vector in which ID(1)=1. For example, MR=4 for (0,1,1,0,0,0), and IST=5 for (1,0,1,0,0,0,1).

The first fathoming test determines whether the current partial solution is complete or not. If MR=1, the partial solution is complete, and therefore, a backward move should be made by identifying IST. If IST is greater than NP, the run terminates because each element in the solution vector to be considered is zero. If the current partial solution passes the first fathoming tests, the second fathoming test is performed by calling the subroutine FEASI(ID). If PI(NP) is greater than PAOQ, a backward move is made by identifying

MR or IST. Procedures for generating a new partial solution by backward move are illustrated in the flowchart in Appendix C.2. If PI(NP) is less than or equal to PAOQ, the third fathoming test is performed by calling ZTEST (ID,IDPR,ZC). If ZC is greater than ZMIN, a backward move is made. If not, ZMIN and IDMI(I) are updated, and a forward move is made to generate a new partial solution by identifying MR.

The procedure for converting final IDMI(I), and the format of the output are exactly the same as those of the program LAWLER.

Descriptions of Computer Output

A nonserial production-inspection system as shown in Figure 5-4 is analyzed by Glover's enumeration scheme according to the procedure described in Appendix D. The program GLOVER has been saved in the permanent file GLOVER while all necessary input data for the system in Figure 5-4 have been saved in the file Y10BH.

The resulting computer output is illustrated in Table 4-2 which includes values of all constants as well as the optimal solution obtained.

The column I in Table 4-2 represents the number of each operation or inspection station, and the columns PA, PB, PE, CL, CM, CS, and CV include values of a_i, b_i, e_i, l_i, c_i, s_i, and v_i, respectively. The expected external failure costs (CK), the total number of the final product (TOUT), and the desired average outgoing quality of the final product (P_{AOQ}) are also included.

Table 4-2. A Sample Computer Output

ANALYSIS ØF NØNSERIAL PRØDUCTIØN-INSPECTIØN SYSTEMS
BY GLØVERS SCHEME

** INPUT INFØRMATIØN

NØ. ØF ØPS = 10 NØ. ØF SØURCES = 2

I	PA	PB	PE	CL	CM	CS	CV
1	.08	.08	.05	.08	6.20	6.20	3.72
2	.03	.07	.09	.14	3.70	3.70	2.22
3	.01	.04	.08	.21	2.60	16.20	9.72
4	.09	.05	.07	.26	3.80	20.00	12.00
5	.07	.05	.01	.29	7.10	59.50	35.70
6	.09	.08	.02	.31	9.30	29.30	17.58
7	.01	.04	.05	.39	6.40	84.50	50.70
8	.01	.05	.09	.43	8.30	37.60	22.56
9	.09	.07	.03	.52	8.00	289.80	173.88
10	.03	.02	.08	.55	6.80	382.90	229.74

CK = 229.740 TØUT = 1.000 PAØQ = .010

** ØPTIMAL SØLUTIØN

MINIMUM CØST = 79.342172

I	D	PØ	PI
1	1	.050000	.004000
2	1	.090000	.002700
3	1	.088621	.000886
4	1	.070824	.006374
5	0	.018053	.018053
6	0	.026247	.026247
7	1	.078300	.000783
8	1	.113884	.001139
9	0	.034824	.034824
10	1	.121216	.003636

TØTAL NØ. ØF VECTØRS ENUMERATED = 240
3.789 CP SECØNDS EXECUTIØN TIME

The column D in the section of OPTIMAL SOLUTION represents the optimal screening plan obtained for the system. Therefore, it is optimal to perform screening inspection at the 1st, 2nd, 3rd, 4th, 7th, 8th, and 10th inspection station within the system in Figure 5-4. The column PO includes the quality level of the product immediately after each operation while the column PI includes the quality level of the product immediately after each inspection station. Therefore, PI(10) represents the average outgoing quality of the final product. Besides, the total quality costs needed to maintain the optimal screening plan are also included.

V. EXPERIMENTAL ANALYSIS

Computational Efficiencies of the Proposed Algorithms

A series of test problems have been generated and solved by the two algorithms proposed in Chapter IV in order to determine which is more efficient for solving the problem formulated in Chapter III.

Test problems have been generated through the following procedures.

1. Three levels of the total number of operations, n, in a nonserial production system are considered (5, 10, and 15).
2. For each n, the constant parameters, a_i, b_i, e_i, l_i, and c_i for i=1,2,...,n are randomly generated from a random number table. For example, each a_i, b_i, and e_i is randomly selected in the interval (0.00,0.10) while each l_i and c_i is randomly selected in the interval (0.00,1.00) and (1.0,10.0), respectively. The salvage value v_i at the i^{th} inspection station is arbitrary fixed at 60% of the cumulative manufacturing costs s_i. The expected external failure costs k is set to 60% of the final cumulative manufacturing costs s_n.
3. The total number of raw material suppliers, m, is fixed at 1, 2, and 3 for n=5, 10, and 15, respectively.
4. For each n, three types of nonserial structures have been generated in such a manner that the density of the flow-rate matrix for a type of structure is different from the others'. The density of a flow-rate matrix is the ratio

of the total number of nonzero elements to the total number of elements in the matrix. Structure A has the lowest density while Structure C has the highest density. The density of Structure B lies between the two. Appendix E shows the flow-rate matricies of A, B, and C for each n.

5. For each n and for each structure, two different values of P_{AOQ} are assigned. The lower is 0.05 while the higher is 0.01.

A test problem generated through the above procedures can be represented by a symbolic name. For example, the problem 10BL is said to have 10 operations (2 raw material suppliers) with a physical structure of type B, and P_{AOQ} is set to 0.05 in that problem. Thus, different test problems have been generated, and each is solved by the two proposed algorithms using the CDC 6400 computer. The total number of vectors enumerated explicitly (NOEN) and the CPU time spent for solving each problem are measured and used for evaluating the computational efficiencies of the two algorithms. The summarized results are shown in Table 5-1, which is rearranged as shown in Table 5-2, 5-3, and 5-4 for better comparison between the two algorithms.

The mean CPU Time (MCPU) and mean NOEN (MNOEN) related to each algorithm are shown in Table 5-2 for each problem. The relationship between MCPU and n is shown in Figure 5-1. Clearly, MCPU as well as MNOEN for both algorithms increases rapidly as n increases. However, Glover's scheme always gives less MCPU and MNOEN than Lawler and Bell's algorithm, and this trend becomes more evident as n increases.

Table 5-1. Experimental Results for Test Problems.

Problem	Lawler and Bell's Algorithm CPU Time (sec)	Lawler and Bell's Algorithm NOEN	Glover's Scheme CPU Time (sec)	Glover's Scheme NOEN
5AL	.210	25	.169	24
5AH	.135	13	.113	12
5BL	.168	19	.144	19
5BH	.103	9	.099	9
5CL	.149	15	.133	15
5CH	.088	6	.081	6
10AL	8.317	399	6.655	390
10AH	6.673	327	5.230	318
10BL	7.791	366	6.467	367
10BH	4.564	246	3.794	240
10CL	7.267	336	5.938	334
10CH	3.135	159	2.676	156
15AL	589.032	13,621	473.951	12,944
15AH	354.699	8,231	269.702	7,287
15BL	510.817	11,506	405.953	11,366
15BH	243.310	6,852	224.200	6,543
15CL	353.337	8,048	321.885	8,482
15CH	73.274	2,002	72,506	2,063

Table 5-2. Mean CPU Time and Mean NOEN for Problems with n Operations.

n	Lawler and Bell's Algorithm MCPU (sec)	Lawler and Bell's Algorithm MNOEN	Glover's Scheme MCPU (sec)	Glover's Scheme MNOEN
5	.142	14.5	.123	14.2
10	6.291	305.5	5.127	300.8
15	354.078	8376.7	294.700	8114.2

Table 5-3 shows MCPU and MNOEN for each algorithm for varying n and P_{AOQ}. Figure 5-2 represents the relationships between MCPU and P_{AOQ} for each n. For convenience, common logarithms of MCPU are taken and plotted in Figure 5-2. Note that problems with P_{AOQ}

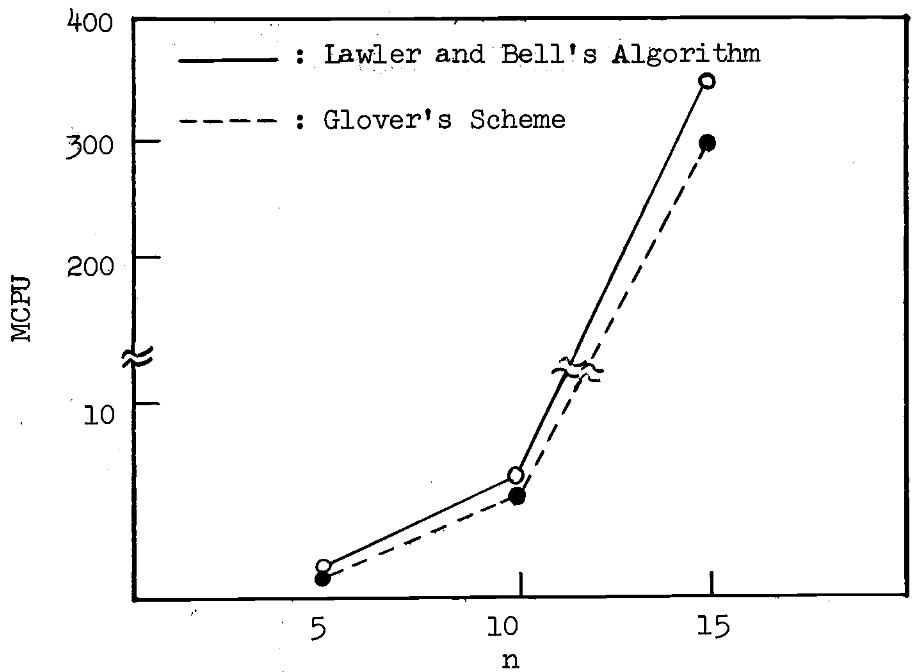

Figure 5-1. Mean CPU Time versus n.

Table 5-3. Mean CPU Time and Mean NOEN for Problems with Different Levels of P_{AOQ}.

n	P_{AOQ}	Lawler and Bell's Algorithm		Glover's Scheme	
		MCPU (sec)	MNOEN	MCPU (sec)	MNOEN
5	.05	.176	19.7	.149	19.3
	.01	.109	9.3	.098	9.0
10	.05	7.792	367.0	6.353	367.7
	.01	4.791	244.0	3.900	238.0
15	.05	484.395	11058.3	400.596	10930.7
	.01	223.761	5695.0	188.803	5297.7

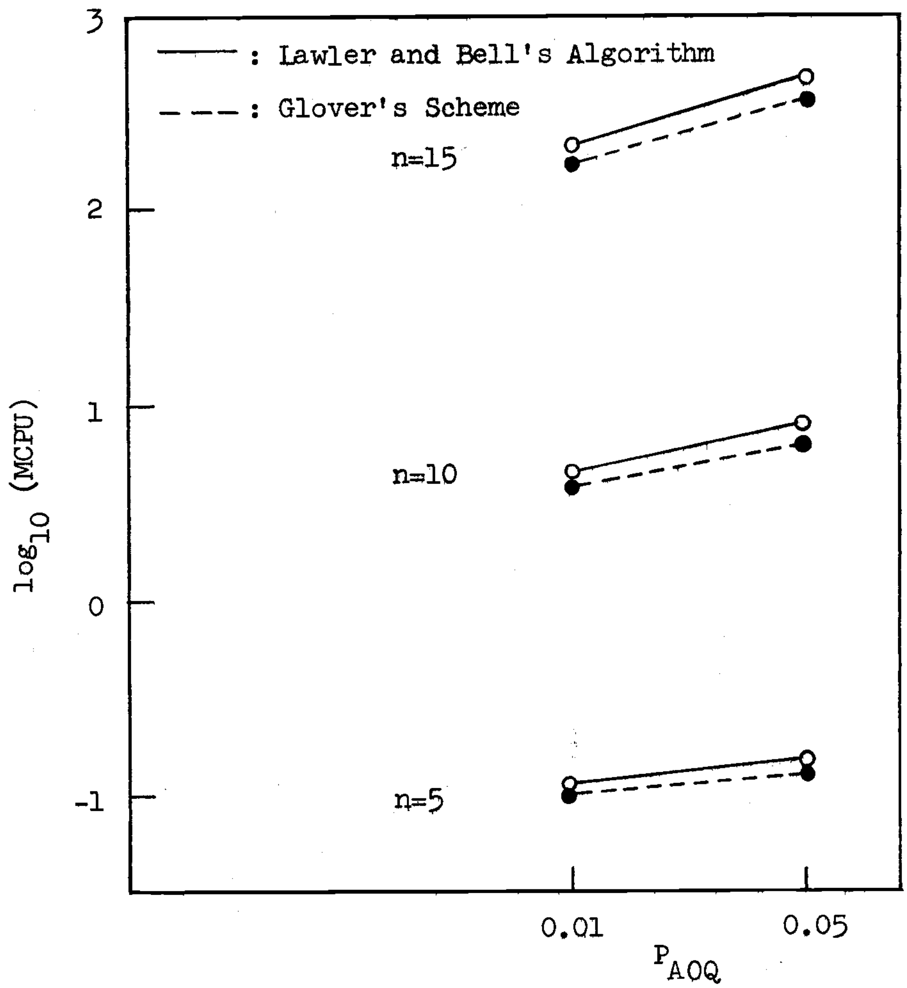

Figure 5-2. Mean CPU Time versus P_{AOQ}.

of 0.01 takes less CPU time than problems with P_{AOQ} of 0.05 for each n.

Table 5-4 shows MCPU and MNOEN for each algorithm with respect to n and types of physical structures. As mentioned earlier, structure C has the highest density of flow-rate matrix while structure A has the lowest. The relationships between MCPU and physical structures are illustrated in Figure 5-3. The values on the vertical axis in Figure 5-3 are common logarithms of MCPU. It is of interest to note that problems with higher density need less computation time than problems with lower density for each n, and that the rate of decrease appears to increase with increasing n.

Table 5-4. Mean CPU Time for Problems with Different Types of Physical Structures.

n	Type of Structure	Lawler and Bell's Algorithm		Glover's Scheme	
		MCPU	MNOEN	MCPU	MNOEN
5	A	.173	19	.141	18
	B	.136	14	.122	14
	C	.119	10.5	.107	10.5
10	A	7.495	363	5.943	354
	B	6.178	306	5.131	303.5
	C	5.201	247.5	4.307	245
15	A	471.866	10926	371.827	10115.5
	B	377.064	9179	315.077	8954.5
	C	213.306	5025	197.196	5272.5

Based upon the above results the following conclusions can be drawn regarding the computational efficiency of the algorithms.

1. Computation time needed to solve a problem increases rapidly as the problem size increases for both algorithms. This is because the total number of solution points to be enumerated increases exponentially as the problem size increases.

2. Glover's enumeration scheme is better than Lawler and Bell's algorithm for solving the formulated problem in the sense that it requires less computation time. As mentioned in Chapter IV, the procedure for generating d'*-1 in Lawler

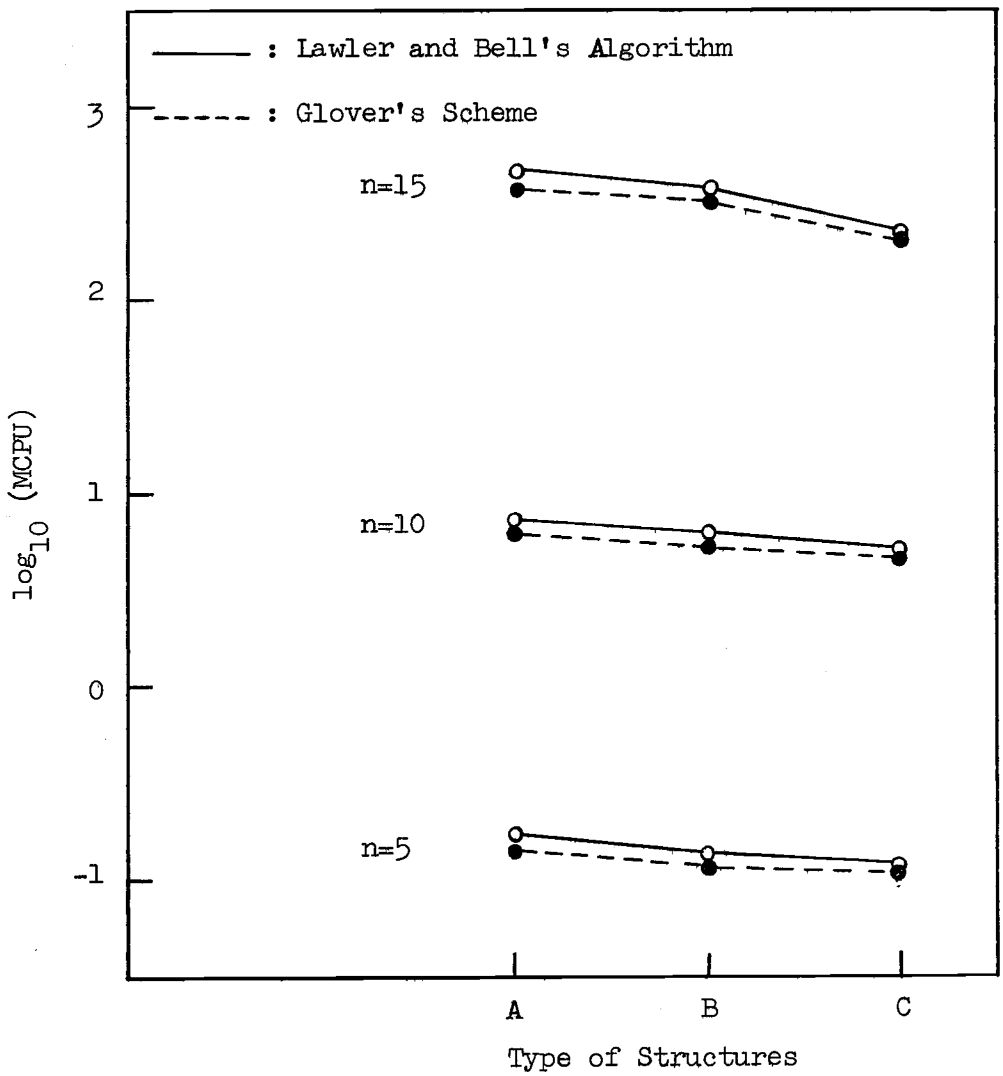

Figure 5-3. Mean CPU Time versus Type of Physical Structures.

and Bell's algorithm is more complicated than the procedure for generating (d'-)*-1 in Glover's scheme. It is believed that this is the reason why Glover's scheme requires less computation time.

3. A problem with smaller P_{AOQ} takes less computation time than the problem with greater P_{AOQ} because the total number

of vectors explicitly enumerated decreases as P_{AOQ} decreases. In other words, more solution points are discarded by the fathoming tests related with the feasibility of the problem as P_{AOQ} becomes smaller.

4. A problem with higher density of flow-rate matrix takes less computation time than the problem with lower density of flow-rate matrix. It is believed that this result is also related with the fathoming tests related with the feasibility of the problem. As the density of flow-rate matrix becomes higher there exist more interactions between operations, and these interactions affect the quality of the final product. Therefore, more solution points can be discarded by the fathoming tests as the density of flow-rate matrix becomes higher.

Analysis of Nonserial Production-Inspection Systems

After finding an optimal solution to the initial problem, a decision maker commonly wants to do additional computations to obtain more information about the problem just solved. The need for this postoptimal analysis arises due to variations of constants as well as uncertainties involved in obtaining their estimates. For example, process averages are subject to change due to natural and assignable causes. A decision maker may want to know the effect of those variations on the optimal solution obtained. Another example is that the external failure costs may not be known with certainty as discussed in Chapter I, so that, the decision maker must estimate them to solve the problem. In this case, he can perform a sensitivity analysis to determine the range of the external failure costs over which a solution remains optimal. If the results show that the optimal solution is quite sensitive to certain values of the external failure costs, the decision maker should spend additional effort in order to obtain better estimates of these costs.

Based upon the above discussions, two types of postoptimal analyses are performed for a given production-inspection system.

Analysis 1

Suppose that a constant in a given system is likely to be changed, and then, it is desirable to analyze the effect of those variations on the optimal solution as well as on the total quality costs and the average outgoing quality level. In this case, the purpose can be achieved by solving a sequence of problems in which the varying constant is gradually increased while all the other constants are fixed at certain values.

For the convenience of the analysis a system with 10 operations is chosen and it is assumed to have a type B physical structure as illustrated in Figure 5-4. It is also assumed the expected external failure costs and P_{AOQ} are fixed at 229.74 and 0.1, respectively. The other constants are fixed as shown in Appendix E.1 except e_1 which varies from 0.01 to 0.2. For this sensitivity analysis, the increment of e_1 is set to 0.01, and a sequence of problems are solved by Glover's enumeration scheme for each e_1. After finding

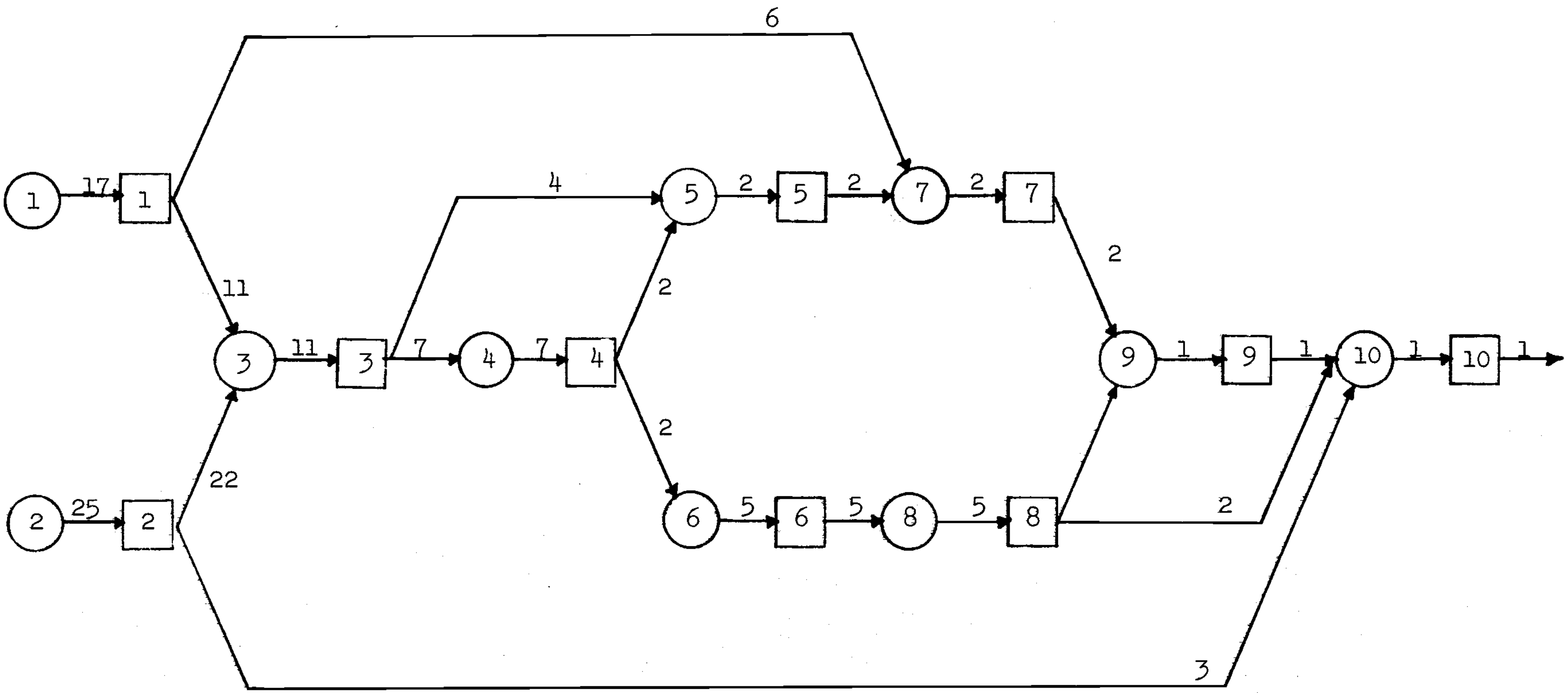

Figure 5-4. A Nonserial Production-Inspection System for Analysis 1 and 2.

an optimal solution for each problem, a range in doubt is investigated again by setting the increment of e_1 to 0.005. Table 5-5 shows the results of this analysis. Note that the optimal solution is changed immediately when e_1 reaches to 0.025, and remains unchanged after that point. This critical point can be approximately determined by trial and error. A more accurate value of the critical point can be achieved by evaluating several points in the interval (0.02, 0.025). The total quality costs per unit of final product are increased as e_1 is increased, however, the actual average outgoing quality level of final product remains almost unchanged regardless of variations in e_1, because of the constraint on the P_{AOQ}.

Using these results, a decision maker can take several alternate courses of action. If e_1 is greater than or equal to 0.025, his optimal action is to perform incoming inspection at the first inspection station. Besides, the increase in the total quality costs due to the increase in e_1 may be compensated by discounts in purchasing costs of raw materials. Similar analyses can be performed for the problems in which other constants are subject to change.

Table 5-5. The Results of Analysis 1. *

Quality of the First Raw Materials (e_1)	Optimal Solution $d=(d_{10},d_9\ldots,d_1)$	Total Quality Costs	Actual AOQL
.01	(1,0,1,1,0,0,1,1,1,0)	74.33	.0036
.02	(1,0,1,1,0,0,1,1,1,0)	76.78	.0037
.025	(1,0,1,1,0,0,1,1,1,1)	77.96	.0036
.05	(1,0,1,1,0,0,1,1,1,1)	79.34	.0036
.1	(1,0,1,1,0,0,1,1,1,1)	82.11	.0036
.2	(1,0,1,1,0,0,1,1,1,1)	87.62	.0037

* The external failure costs and P_{AOQ} are fixed at 229.74 and 0.1, respectively.

Analysis 2

This analysis is related with problems in which a constant is subject to uncertainty. Suppose that a decision maker can only obtain a rough estimate of a constant, and therefore, he cannot be certain whether the optimal solution is truly optimal or not. In this case, the best alternative that he can take is to perform a sensitivity analysis with respect to the uncertain constant.

As an example, the same system as in the Analysis 1 is chosen with P_{AOQ} being fixed at 0.2. It is assumed that the expected external failure costs are uncertain. All the other constants are assumed to be fixed as shown in Appendix E.1, and a sequence of problems in which the external failure costs vary from 0 to 400 is solved. The summarized results are shown in Table 5-6 in which two critical points, 37.5 and 190, can be identified. These points can be determined approximately as mentioned in the Analysis 1.

Table 5-6. The Results of Analysis 2.*

Expected External Failure Costs (CK)	Optimal Solution $d=(d_{10},d_9,\ldots,d_1)$	Total Quality Costs	Actual AOQL
0.0	(0,0,1,1,0,0,0,1,1,1)	57.13	.125
5.0	(0,0,1,1,0,0,0,1,1,1)	57.75	.125
10.0	(0,0,1,1,0,0,0,1,1,1)	58.38	.125
30.0	(0,0,1,1,0,0,0,1,1,1)	60.87	.125
37.5	(0,0,1,1,0,0,1,1,1,1)	61.80	.121
40.0	(0,0,1,1,0,0,1,1,1,1)	62.11	.121
100.0	(0,0,1,1,0,0,1,1,1,1)	69.38	.121
180.00	(0,0,1,1,0,0,1,1,1,1)	79.07	.121
190.0	(1,0,1,1,0,0,1,1,1,1)	79.20	.004
300.0	(1,0,1,1,0,0,1,1,1,1)	79.60	.004
400.0	(1,0,1,1,0,0,1,1,1,1)	79.96	.004

* P_{AOQ} is fixed at 0.2.

Using these results, a decision maker can determine the range of the external failure costs over which an optimal solution remains unchanged. Note that the optimal solution is not quite sensitive to the change in the external failure costs, in other words, each range is fairly broad. Therefore, although a decision maker cannot determine the precise value of the external failure costs, he can choose an optimal policy with certainty as long as he is certain that the external failure costs lie over the related range. Besides, the actual P_{AOQ} is plotted as a function of the external failure costs as shown in Figure 5-5.

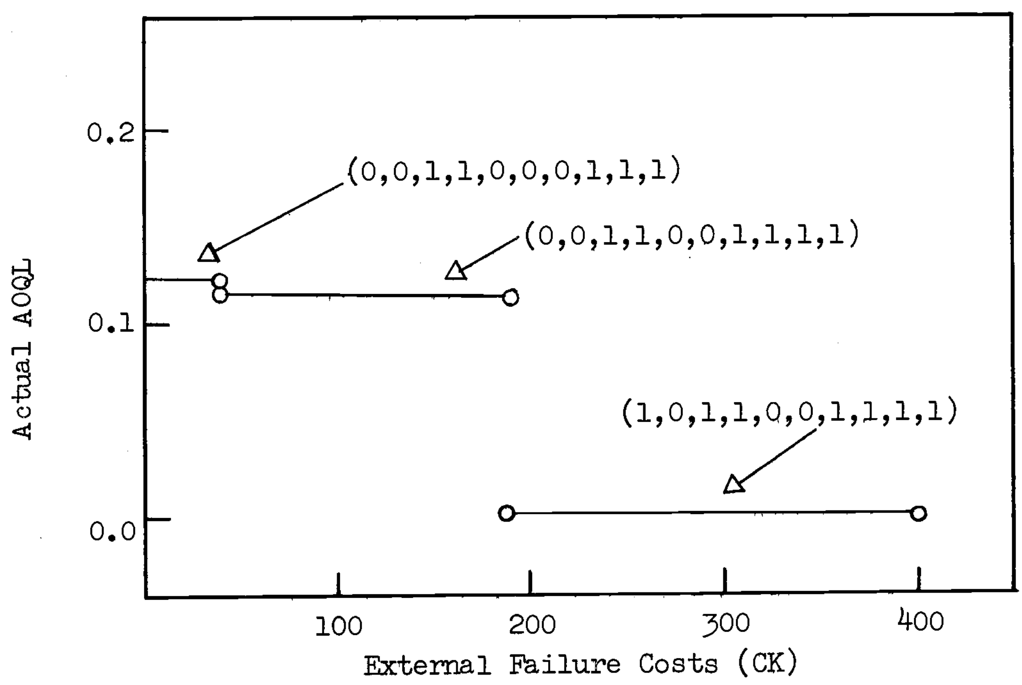

Figure 5-5. Actual AOQL and Optimal Solutions for Varying External Failure Costs.

VI. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Summary

The economic aspect of nonserial production-inspection systems has been systematically investigated. First, the problem was defined as how to determine optimal location of screening inspection stations within a nonserial production process. In addition, various aspects of nonserial production-inspection systems were introduced in order to evaluate their effects on the problem solution. Then current approaches to the problem were surveyed and their deficiencies were identified. For instance, most of the current approaches have ignored the constraints existing in general production-inspection systems as well as the effect of imperfect inspection on the problem.

Realizing the above deficiencies, a mathematical model was developed for solving the problem under the assumption of imperfect inspection. The model was formulated as a constrained nonlinear zero-one integer programming problem in which the objective function represents the total quality costs needed to maintain a screening inspection plan, the constraint represents the requirement of the average outgoing quality of the final product, and a binary variable represents the decision made at each inspection station. In addition, it was shown that the model can easily reflect various limitations and requirements of production-inspection systems.

It was then shown that the objective function as well as the constraint in the formulated problem is monotone nondecreasing in each of decision variables. This unique mathematical structure of the problem was used for providing fathoming tests for proposed solution techniques.

Lawler and Bell's algorithm and Glover's enumeration scheme were proposed as suitable solution techniques to the formulated problem. The two algorithms were programmed and their computational efficiencies were compared with each other by solving randomly generated test problems. Each test problem has different physical structure as well as different values of n, m, and P_{AOQ} from others.

Finally, as an application of the model, some postoptimal analysis was performed for a nonserial production-inspection systems.

Conclusions

It is believed that the mathematical model developed has several advantages compared to other models currently available. First, the model is flexible enough to reflect various limitations and requirements existing in general production-inspection systems. For example, the requirement of the average outgoing quality of the final product, the availability of inspection effort, physical difficulties for allocating inspection at certain points, etc. can be easily incorporated into the model as discussed in Chapter III. Moreover, the model can be applied to a nonserial production-inspection system in which different final products are produced from different final operations. Secondly, it is believed that the model is more realistic than others in the sense that it deals with nonserial production systems including serial ones as special cases, and that imperfect inspection is explicitly considered. Thirdly, the procedures for building the model is quite straightforward, and therefore, no profound mathematical background is needed.

In addition to these advantages, several important facts have been discovered by establishing a theorem in Chapter III. According to the theorem the product quality immediately after each operation or inspection station is monotone nondecreasing in each of decision variables $d'_1, d'_2, \ldots, d'_n$. Using this theorem it was proved that the objective function can be expressed as the difference of two monotone nondecreasing functions, and that the constraint is also monotone nondecreasing in each of decision variables. Moreover, this unique mathematical structure of the model remains unchanged after incorporating various limitations and requirements of production-inspection systems into the model.

Of the two solution techniques proposed for solving our constrained nonlinear zero-one integer programming problem, Glover's enumeration scheme was found to be the most efficient under the given conditions for generating test problems. The reason for this result is believed that the procedure for generating next solution point to be enumerated in Glover's scheme is simpler than that of Lawler and Bell's algorithm. However, it was found that the computation time needed to solve the problem grows exponentially as the problem size increases. However, it seems that such computational deficiencies are not limited to our own problem because the computational aspects of current algorithms for solving nonlinear and/or integer programs are not satisfactory (Taha, 1975).

Recommendations

It is recommended that additional research is desired to reduce the computation time required to solve a problem of realistic size. There may be several alternatives to be considered. The first alternative is to provide stronger fathoming tests in order to accelerate the partial enumeration by extensively analyzing the mathematical structure of the problem. The second alternative is to improve time-consuming calculations in the solution procedures. For example, the process for calculating P_j and P'_j for a solution point is most time-consuming, and therefore, an improvement in these calculations will result in great savings in the overall computation time.

Besides, it may be useful to expand the model to the nonserial production-inspection system in which sampling inspection is performed at each potential inspection station.

BIBLIOGRAPHY

Beightler, C. S. and Mitten, L. G. 1964. Design of an Optimal Sequence of Interrelated Sampling Plans. Journal of the American Statistical Association, Vol. 59, pp. 96-104.

Britney, R. R. 1972. Optimal Screening Plans for Nonserial Production Systems. Management Science, Vol. 18, pp. 550-559.

Brunk, H. D. 1975. An Introduction to Mathematical Statistics, 3rd ed. Xerox Corporation, Toronto.

Case, K. E., Bennett, G. K. and Schmidt, J. W. 1975. The Effect of Inspection Error on Average Outgoing Quality. Journal of Quality Technology, Vol. 7, pp. 28-33.

Dietrich, D. L. and Sanders, J. L. 1974. A Bayesian Quality Assurance Model for a Multistage Production Process. ASQC Technical Conference Transactions-Boston, pp. 338-348.

Eppen, G. D. and Hurst, E. G. 1974. Optimal Location of Inspection Stations in a Multistage Production Process. Management Science, Vol. 20, pp. 1194-1200.

Feller, W. 1968. An Introduction to Probability Theory and Its Applications, Vol. I, 3rd ed. John Wiley & Sons, New York.

Fruehwirth, M. Z. 1970. An Economic Criterion for Minimizing Overall Inspection and Repair Cost. Western Electric Engineer, Vol. 14, pp. 27-32.

Garfinkel, R. S. and Nemhauser, G. L. 1972. Integer Programming. John Wiley & Sons, New York.

Geoffrion, A. M. and Marsten, R. E. 1972. Integer Programming: A Frame Work and State-of-the-art Survey. Management Science, Vol. 18, pp. 465-491.

Glover, F. 1965. A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem. Operations Research, Vol. 13, pp. 879-919.

Groocock, J. M. 1974. The Cost of Quality. Pitman Publishing Corporation, New York.

Hurst, E. G. 1973. Imperfect Inspection in a Multistage Production Process. Management Science, Vol. 20, pp. 378-384.

Juran, J. M. and Gryna, F. M. 1970. Quality Planning and Analysis. McGraw-Hill Book Company, New York.

Juran, J. M. (ed.) 1974. Quality Control Handbook, 3rd ed. McGraw-Hill Book Company, New York.

Kirkpatrick, E. G. 1970. Quality Control for Managers and Engineers. John Wiley & Sons, New York.

Lawler, E. L. and Bell, M. D. 1966. A Method for Solving Discrete Optimization Problems. Operations Research, Vol. 14, pp. 1098-1112.

Lindsay, G. F. and Bishop, A. B. 1964. Allocation of Screening Inspection Effort: A Dynamic Programming Approach. Management Science, Vol. 10, pp. 342-352.

Lindsay, G. F. 1967. A Dynamic Programming Procedure for Locating Inspection Stations. AIIE Proceedings, pp. 244-248.

Pruzan, P. M. and Jackson, J. T. 1967. A Dynamic Programming Application in Production Line Inspection. Technometrics, Vol. 9, pp. 73-81.

Riggs, J. L. 1976. Production Systems: Planning, Analysis, and Control, 2nd ed., John Wiley & Sons, New York.

Salkin, H. M. 1975. Integer Programming. Addison-Wesley Publishing Company, Reading, Mass.

Taha, H. A. 1975. Integer Programming: Theory, Applications, and Computations. Academic Press, New York.

Trippi, R. R. 1974. An On-Line Computational Model for Inspection Resource Allocation. Journal of Quality Technology, Vol. 6, pp. 167-174.

Trippi, R. R. 1975. The Warehouse Location Formulation as a Special Type of Inspection Problem. Management Science, Vol. 21, pp. 986-988.

Western Electric Company. 1958. Statistical Quality Control Handbook, 2nd ed., New York.

White, L. S. 1966. The Analysis of a Single Class of Multistage Inspection Plans. Management Science, Vol. 12, pp. 685-693.

White, L. S. 1969. Shortest Route Modes for the Allocation of Inspection Effort on a Production Line. Management Science, Vol. 15, pp. 249-259.

Woo, W. K. and Metcalfe, J. E. 1972. Optimal Allocation of Inspection Effort in Multi-stage Manufacturing Processes. Western Electric Engineers, Vol. 16, pp. 8-16.

Zangwill, W. 1969. Nonlinear Programming: A Unified Approach. Prentice Hall, Englewood Cliffs, N. J.

APPENDICES

APPENDIX A. Symbols Used in the Mathematical Model.

Symbol	Definition
a_j	Pr (Type II errors occur at the j^{th} inspection station.)
b_j	Pr (Type I errors occur at the j^{th} inspection station.)
c_j	Machining costs per unit at the j^{th} operation.
d_j	A decision variable at the j^{th} inspection station. (Equals 1 if inspection is performed at the j^{th} inspection station, and equals 0 otherwise).
d'_j	Equals $1-d_j$.
e_j	Pr (Operational errors occur at the j^{th} operation)
k	Expected external failure costs per unit of final product.
l_j	Sum of apprisal costs per unit of the j^{th} product.
m	Total number of raw material suppliers.
n	Total number of operations or total number of potential inspection stations.
P_{AOQ}	Required Average Outgoing Quality Level.
P_j	Pr (A unit of product immediately after the j^{th} operation has at least one type of defect)
P'_j	Pr (A unit of product immediately after the j^{th} inspection station has at least one type of defect)
R	Flow-rate matrix for a nonserial production system.
r_{ij}	Material flow rate from the i^{th} inspection station to the j^{th} operation to achieve one unit of the final product $(i \neq j)$.
	Material flow rate from the i^{th} operation to the j^{th} inspection station $(i = j)$.
s_j	Cumulative manufacturing costs per unit of product immediately after the j^{th} operation.
v_j	Salvage value per unit of the j^{th} product.

APPENDIX B. A FORTRAN Program for Lawler and Bell's Algorithm.

B.1. Symbolic Names Used in the Program LAWLER.

Group	Symbol	Definition
Variable	ID(I)	A solution vector currently considered.
	IDPR(I)	An array for representing d'*-1.
	IDMI(I)	An optimal solution vector so far attained.
	PO(I)	An array for representing P_i, i=1,2,...,n.
	PI(I)	An array for representing P'_i, i=1,2,...,n.
	PONT(I)	PO(I)
	PINT	PI(NP)
	ZMIN	An optimal value of the objective function so far attained.
	NOEN	Total number of solution vectors explicitly enumerated.
Constant	RATE(I,J)	A two-dimensional array for representing material flow rates ($=r_{ij}$).
	PA(I)	An array for representing probabilities of Type II errors ($=a_i$).
	PB(I)	An array for representing probabilities of Type I errors ($=b_i$).
	PE(I)	An array for representing probabilities of operational errors ($=e_i$).
	CL(I)	An array for representing appraisal costs per unit inspected ($=l_i$).
	CM(I)	An array for representing machining costs per unit processed ($=c_i$).
	CS(I)	An array for representing cumulative manufacturing costs per unit processed ($=s_i$).
	CK	Expected external failure costs per unit detected ($=k$).

APPENDIX B.1. (Continued)

Group	Symbol	Definition
Constant	NP	Total number of operations or total number of potential inspection stations (=n).
	IN	Total number of raw material suppliers (=m)
	TOUT	Total amount of final product.
	PAOQ	Desired average outgoing quality of the final product.
Subroutine	OVFLW(IA,M)	Determines whether a solution vector IA reaches to the final vector or not. If M=1, IA reaches to the final vector, and therefore, the run terminates.
	DPLUS(IA)	Generates d'+1 from d'. d' = IA
	DMIUS(IA)	Generates d'-1 from d'. d' = IA
	DSTMI(IA)	Generates d'*-1 from d'. d' = IA
	FEASI(IA)	Calculates P_j and P'_j for j = 1,2,...,n for a solution vector IA.
	OBJCT(IA, ZD)	Calculates the value (=ZD) of the objective function for a solution vector IA.
	ZTEST(IA, IB,Z)	Calculates the value (=Z) of Z_1(IA) - Z_2(IB) for the second fathoming test. IA=d' and IB=d'*-1.
	EQUAL(IA, IB)	Equalize a vector IB to another vector IA.

Appendix B.2. Flow Chart of the Program LAWLER

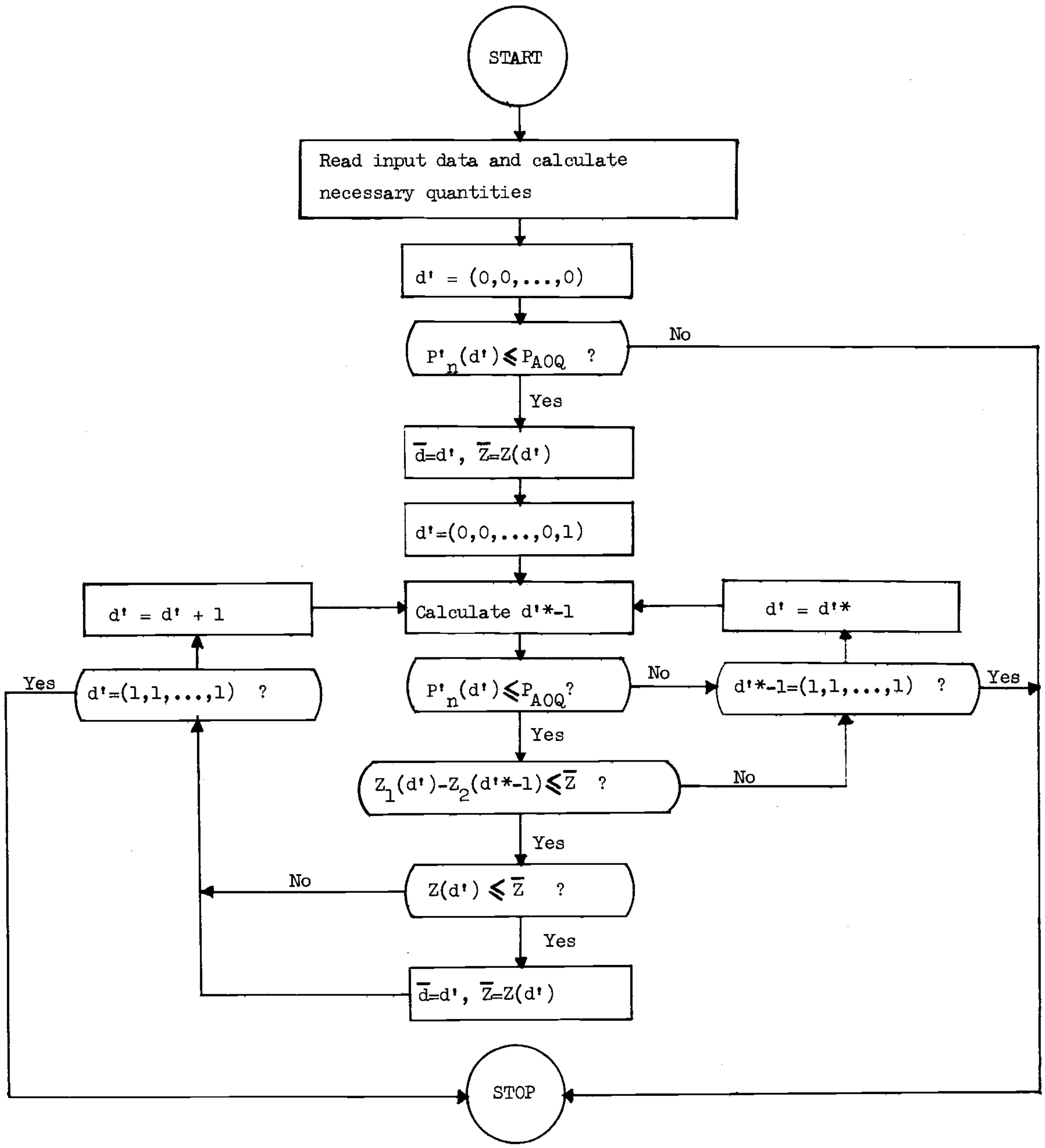

Figure B-1. Flow Chart for Lawler and Bell's Algorithm.

APPENDIX B.3. A Listing of the Program LAWLER.

```
      PROGRAM LAWLER(INPUT,OUTPUT,TAPE60=INPUT,TAPE61=OUTPUT)
C
      DIMENSION ID(50), IDPR(50), IDMI(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOG,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),PINT
C
C     READ INPUT DATA
C
      READ(60,10) NP,IN,TOUT,   PAOG,(PE(I),PA(I),PB(I),CL(I),CM(I),
     1I=1,NP)
      READ(60,20) ((RATE(J,K),K=1,NP),J=1,NP)
C
C     CALCULATE NECESSARY QUANTITIES
C
      DO 15 I=1,IN
      CS(I) = CM(I)
   15 CONTINUE
      IJ=IN+1
      DO 25 J=IJ,NP
      TSUM = 0.
      JM=J-1
      DO 35 L=1, JM
      TSUM = TSUM + CS(L) * ( RATE(L,J)/RATE(J,J))
   35 CONTINUE
      CS(J)=TSUM+CM(J)
   25 CONTINUE
C
      DO 45 I=1,NP
      RL(I)=RATE(I,I)*CL(I)
      CV(I)=0.6*CS(I)
      CE(I)=0.4*CS(I)
      RS(I)=RATE(I,I)*CE(I)
   45 CONTINUE
      CK=0.6*CS(NP)
C
C     START ENUMERATION
C     CREATE INITIAL SOLUTION
C     SET UP INITIAL UPPER BOUND
C
      DO 100 I = 1, NP
      ID(I) = 0
  100 CONTINUE
C
```

```
C      APPLY FATHOMING TESTS
C
       CALL FEASI(ID)
       IF(PI(NP) .GT. PAOG) GO TO 99
       CALL OBJCT(ID, ZD)
       ZMIN = ZD
       CALL EQUAL(ID, IDMI)
       CALL DPLUS(ID)
C
       NOEN=1
C
   17  CALL EQUAL(ID,IDPR)
       CALL DSTMI(IDPR)
       CALL FEASI(ID)
       NOEN=NOEN+1
       IF(PI(NP).GT.PAOG) GO TO 27
       CALL ZTEST(ID,IDPR,Z)
       IF(Z.GT.ZMIN) GO TO 27
       DO 300 I=1,NP
       PO(I)=PONT(I)
  300  CONTINUE
       PI(NP)=PINT
       CALL OBJCT(ID,ZD)
       IF(ZD.GE.ZMIN) GO TO 37
       ZMIN=ZD
       CALL EQUAL(ID,IDMI)
C
C      APPLY THE STOPPING RULE
C
   37  CALL OVFLW(ID,M1)
       IF(M1.EQ.1) GO TO 800
       CALL DPLUS(ID)
       GO TO 17
C
   27  CALL OVFLW(IDPR,M2)
       IF(M2.EQ.1) GO TO 800
       CALL DPLUS(IDPR)
       CALL EQUAL(IDPR,ID)
       GO TO 17
C
   99  WRITE(61, 30)
       GO TO 900
C
C      CONVERT THE OPTIMAL SOLUTION
C
  800  CALL FEASI(IDMI)
       DO 200 I = 1, NP
       IDMI(I) = 1 - IDMI(I)
  200  CONTINUE
C
```

```
      WRITE(61,111)
      WRITE(61,40) NP, IN
      WRITE(61,50) (I, PA(I), PB(I), PE(I), CL(I), CM(I), CS(I), CV(I),
     1I=1,NP)
      WRITE(61,70) CK, TOUT, PAOQ
      WRITE(61,112)
      WRITE(61,90) ZMIN,(I,IDMI(I),PO(I),PI(I),I=1,NP)
      WRITE(61,110) NOEN
   10 FORMAT(2I2,2F10.3/(10F5.2))
   20 FORMAT(10F2.0)
   30 FORMAT(1H1,///,5X,"INFEASIBLE")
  111 FORMAT(1H1,//// ,5X,"ANALYSIS OF NONSERIAL PRODUCTION-INSPECTION
     1SYSTEMS",/,5X,"BY GLOVERS SCHEME",///,5X,"** INPUT INFORMATION")
  112 FORMAT(//,5X,"** OPTIMAL SOLUTION")
   40 FORMAT(/,5X,"NO. OF OPS = ",I5,10X,"NO. OF SOURCES = ",I5)
   50 FORMAT(//,6X,"I",5X,"PA",6X,"PB",6X,"PE",6X,"CL",5X,"CM",
     16X,"CS",6X,"CV",//,(5X,I2,7F8.2))
   70 FORMAT(//,5X,"CK = ",F10.3,5X,"IOUT = ",F10.3,5X,"PAOQ = ",F10.3)
   90 FORMAT(/ ,5X,"MINIMUM COST = ",F15.6,//,7X,"I",4X,"D",8X,"PO",8X,
     1"PI",//,(5X,I3,2X,I3,2X,2F10.6))
  110 FORMAT(/ ,5X        ,"TOTAL NO. OF VECTORS ENUMERATED = ",I10)
  900 STOP
      END
```

```
      SUBROUTINE OVFLW(IA,M)
      DIMENSION IA(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOQ,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),PINT
      M=0
      DO 100 I = 1,NP
      K = NP + 1 - I
      IF(IA(K) .EQ. 0) RETURN
  100 CONTINUE
      M = 1
      RETURN
      END
```

```
      SUBROUTINE DPLUS(IA)
      DIMENSION IA(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOQ,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),PINT
      ICAR = 1
      DO 100 I = 1, NP
      IA(I) = IA(I) + ICAR
      IF(IA(I) .NE. 2) RETURN
      IA(I) = 0
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DMIUS(IA)
      DIMENSION IA(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOQ,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),PINT
      ICAR = -1
      DO 100 I = 1, NP
      IA(I) = IA(I) + ICAR
      IF(IA(I) .EQ. 0) RETURN
      IA(I) = 1
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DSIMI(IA)
      DIMENSION IA(50), IB(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOQ,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),PINT
      DO 100 I = 1, NP
      IB(I) = IA(I)
  100 CONTINUE
      CALL DMIUS(IB)
      DO 200 I = 1, NP
      IA(I) = IA(I) + IB(I)
      IF(IA(I) .EQ. 2) IA(I) = 1
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ZTEST(IA, IB, Z)
      DIMENSION IA(50), IB(50), FONT(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOQ,RL(50),RS(50)
     2,CV(50),CE(50),FONT(50),FINT
      TSUM1 = 0.
      TSUM2 = 0.
      DO 100 I = 1, NP
      FONT(I) = PO(I)
  100 CONTINUE
      FINT=PI(NP)
      CALL FEASI(IB)
      DO 200 J = 1, NP
      TSUM1=TSUM1+RL(J)+RS(J)*(PB(J)+FONT(J)+FONT(J)*(PA(J)+PB(J))*
     1IA(J))
      TSUM2=TSUM2+RL(J)*IB(J)+RS(J)*(PO(J)*(PA(J)+PB(J))+IB(J)*
     1(PO(J)+PB(J)))
  200 CONTINUE
      Z=TOUT*(TSUM1-TSUM2+CK*FINT)
      RETURN
      END
```

```
      SUBROUTINE EQUAL(IA, IB)
      DIMENSION IA(50), IB(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOQ,RL(50),RS(50)
     2,CV(50),CE(50)
      DO 100 I = 1, NP
      IB(I) = IA(I)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FEASI(IA)
      DIMENSION IA(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOG,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),PINT
      DO 100 I = 1, IN
      PO(I) = PE(I)
      PI(I) = PO(I) * (PA(I) + (1. - PA(I)) * IA(I))
  100 CONTINUE
      IJ=IN+1
      DO 200 J=IJ,NP
      TMULT = 1.
      JM=J-1
      DO 300 L=1,JM
      TMULT = TMULT * (1. - PI(L)) ** (RATE(L,J) / RATE(J,J))
  300 CONTINUE
      PO(J) = 1. - (1. - PE(J)) * TMULT
      PI(J) = PO(J) * (PA(J) + (1.-PA(J)) * IA(J))
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OBJCT(IA, ZD)
      DIMENSION IA(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOG,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),PINT
      TSUM1 = 0.
      DO 100 I = 1, NP
      TSUM1=TSUM1+(RL(I)+RS(I)*(PB(I)+PO(I)*(1.-PA(I)-PB(I))))*
     1(1.-IA(I))
  100 CONTINUE
      ZD=TOUT*(TSUM1+CK*PI(NP))
      RETURN
      END
```

APPENDIX C. A FORTRAN Program for Modified Glover's Enumeration Scheme.

C.1. Symbolic Names Used in the Program GLOVER.

Group	Symbols	Definition
Variable	ID(I)	A solution vector currently considered.
	IDPR(I)	An array for representing d'+.
	IDMI(I)	An optimal solution vector so far attained.
	PO(I)	An array for representing P_i, i=1,2,...,n.
	PI(I)	An array for representing P'_i, i=1,2,...,n.
	PONT(I)	PO(I).
	PINT	PI(NP).
	ZMIN	An optimal value of the objective function so far attained.
	NOEN	Total number of solution vectors so far enumerated explicitly.
	MR	Indicates the location of the right-most 1 in the current solution vector.
	IST	Indicates the location of the second right-most 1 in a solution vector where ID(NP)= 1. IST is always greater than 1 and less than NP+1.
Constant	RATE(I,J)	A two-dimensional array for representing material flow rates ($=r_{ij}$).
	PA(I)	An array for representing probabilities of Type II error ($=a_i$).
	PB(I)	An array for representing probabilities of Type I error ($=b_i$).
	PE(I)	An array for representing operational errors ($=e_i$).

APPENDIX C.1.(Continued)

Group	Symbol	Definition
Constant	CL(I)	An array for representing appraisal costs per unit inspected ($=l_i$).
	CM(I)	An array for representing machining costs per unit processed ($=c_i$).
	CS(I)	An array for representing cumulative manufacturing costs per unit processed ($=s_i$).
	CK	Expected external failure costs per unit detected (=k).
	NP	Total number of operations or total number of potential inspection stations (=n).
	IN	Total number of raw material suppliers (=m)
	TOUT	Total amount of the final product.
	PAOQ	Desired average outgoing quality of the final product.
Subroutine	DSTMI(IA)	Generates a vector IDPR (=d'+) based upon a vector IA(=d').
	FEASI(IA)	Calculates P_j and P'_j for j=1,2,...,n for a solution vector IA.
	OBJCT(IA, ZD)	Calculates the value (=ZD) of the objective function for a solution vector IA.
	ZTEST(IA, IB,Z)	Calculates the value (=Z) of Z_1(IA)-Z_2(IB) for the second fathoming test. IA=d'- and IB=d'+
	EQUAL(IA, IB)	Equalize a vector IB to another vector IA.

Appendix C.2. Flow Chart of the Program GLOVER

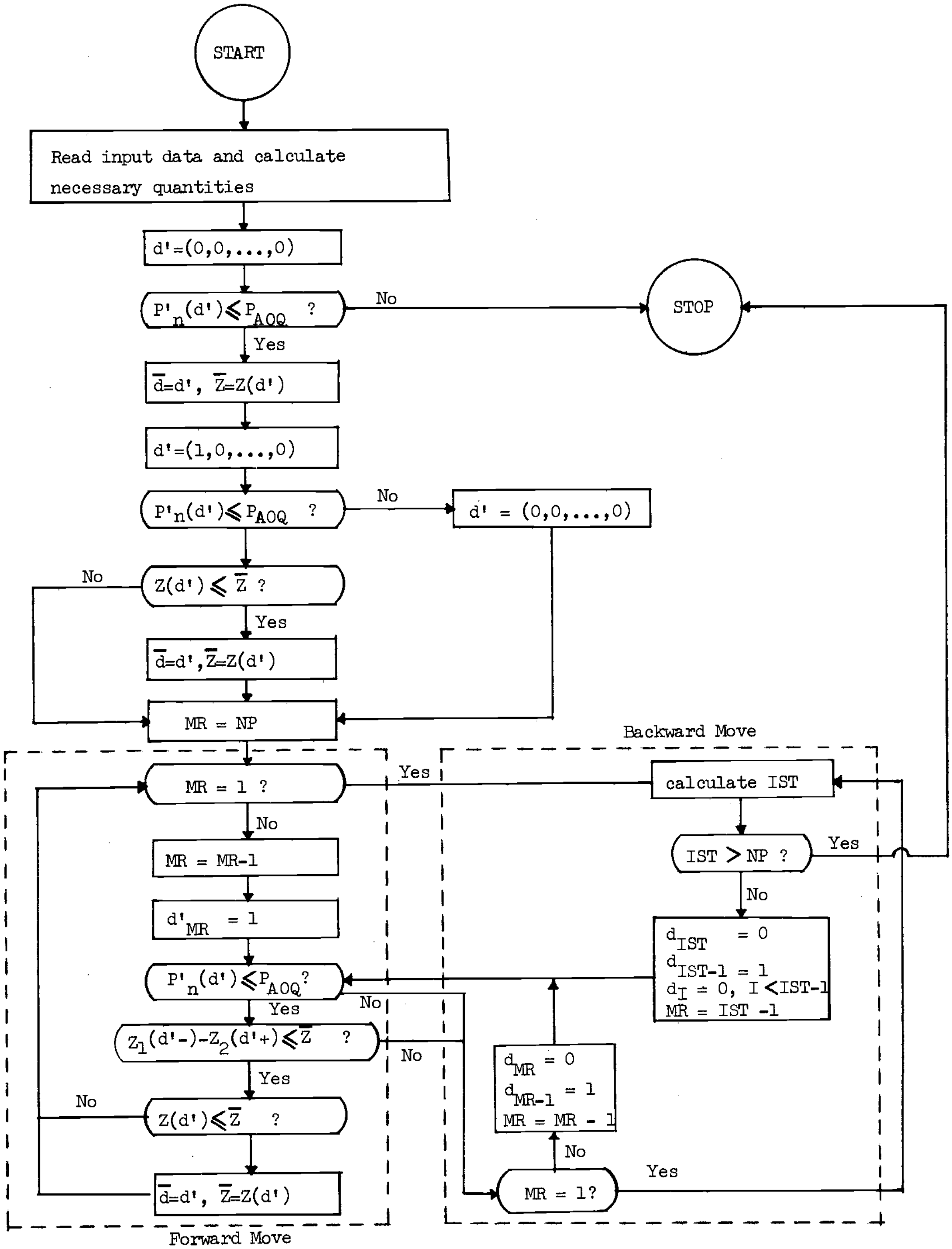

Figure C-1. Flow Chart for Modified Glover's Enumeration Scheme.

APPENDIX B.3. A Listing of the Program GLOVER

```
      PROGRAM GLOVER(INPUT,OUTPUT,TAPE60=INPUT,TAPE61=OUTPUT)
C
      DIMENSION ID(50), IDPR(50), IDMI(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOC,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),MR,IST,PINT
C
C     READ INPUT DATA
C
      READ(60,10) NP,IN,TOUT,   PAOC,(PE(I),PA(I),PB(I),CL(I),CM(I),
     1I=1,NP)
      READ(60,20) ((RATE(J,K),K=1,NP),J=1,NP)
C
C     CALCULATE NECESSARY QUANTITIES
C
      DO 15 I=1,IN
      CS(I) = CM(I)
   15 CONTINUE
      IJ=IN+1
      DO 25 J=IJ,NP
      TSUM = 0.
      JM=J-1
      DO 35 L=1, JM
      TSUM = TSUM + CS(L) * ( RATE(L,J)/RATE(J,J))
   35 CONTINUE
      CS(J)=TSUM+CM(J)
   25 CONTINUE
C
      DO 45 I=1,NP
      RL(I)=RATE(I,I)*CL(I)
      CV(I)=0.6*CS(I)
      CE(I)=0.4*CS(I)
      RS(I)=RATE(I,I)*CE(I)
   45 CONTINUE
      CK=0.6*CS(NP)
C
C     START ENUMERATION
C     CREAT INITIAL SOLUTION
C     SET UP INITIAL UPPER BOUND
C
      DO 100 I = 1, NP
      ID(I) = 0
  100 CONTINUE
C
      NOEN=0
```

```
      CALL FEASI(ID)
      NOEN=NOEN+1
      IF(PI(NP) .GT. PAOQ) GO TO 99
      CALL OBJCT(ID,ZA)
      ZMIN=ZA
      CALL EQUAL(ID, IDMI)
C
C     CREAT ANOTHER SOLUTION BY FORWARD MOVE
C
      ID(NP)=1
      CALL FEASI(ID)
      NOEN=NOEN+1
      IF(PI(NP).GT.PAOQ) GO TO 27
      CALL OBJCT(ID,ZB)
      IF(ZB.GE.ZMIN) GO TO 37
      ZMIN=ZB
      CALL EQUAL(ID,IDMI)
      GO TO 37
   27 ID(NP)=0
   37 MR=NP
C
   17 IF(MR.EQ.1) GO TO 67
      MR=MR-1
      IF(MR.EQ.0) GO TO 67
      ID(MR)=1
C
C     APPLY FATHOMING TESTS
C
   87 CALL FEASI(ID)
      NOEN=NOEN+1
      IF(PI(NP).GT.PAOQ) GO TO 97
      IF(MR.NE.1) GO TO 47
      CALL OBJCT(ID,ZC)
      IF(ZC.GE.ZMIN) GO TO 97
      GO TO 57
   47 CALL EQUAL(ID,IDPR)
      CALL DSTMI(IDPR)
      CALL ZTEST(ID,IDPR,ZC)
      IF(ZC.GE.ZMIN) GO TO 97
      DO 700 I=1,NP
      PO(I)=PONT(I)
  700 CONTINUE
      PI(NP)=PINT
      CALL OBJCT(ID,ZC)
      IF(ZC.GE.ZMIN) GO TO 17
   57 ZMIN=ZC
      CALL EQUAL(ID,IDMI)
      GO TO 17
C
C     IDENTIFY IST AND CREAT ANOTHER SOLUTION BY BACKWARD MOVE
C
   67 DO 300 I=2,NP
      IF(ID(I).EQ.1) GO TO 400
```

```
      IF(I.EQ.NP) GO TO 800
  300 CONTINUE
  400 IST=I
   77 ID(IST)=0
      ID(IST-1)=1
      MR=IST-1
      IF(IST.EQ.2) GO TO 87
      K=MR-1
      DO 600 I=1,K
      ID(I)=0
  600 CONTINUE
      GO TO 87
C
C     IDENTIFY MR AND CREAT ANOTHER SOLUTION BY BACKWARD MOVE
C
   97 IF(MR.EQ.1) GO TO 67
      ID(MR)=0
      ID(MR-1)=1
      MR=MR-1
      GO TO 87
C
   99 WRITE(61, 30)
      GO TO 900
C
C     CONVERT THE OPTIMAL SOLUTION
C
  800 CALL FEASI(IDMI)
      DO 200 I = 1, NP
      IDMI(I) = 1 - IDMI(I)
  200 CONTINUE
C
      WRITE(61,111)
      WRITE(61,40) NP, IN
      WRITE(61,50) (I, PA(I), PB(I), PE(I), CL(I), CM(I), CS(I), CV(I),
     1I=1,NP)
      WRITE(61,70) CK, TOUT, PAOQ
      WRITE(61,112)
      WRITE(61,90) ZMIN,(I,IDMI(I),PO(I),PI(I),I=1,NP)
      WRITE(61,110) NOEN
   10 FORMAT(2I2,2F10.3/(10F5.2))
   20 FORMAT(10F2.0)
   30 FORMAT(1H1,///,5X,"INFEASIBLE")
  111 FORMAT(1H1,//// ,5X,"ANALYSIS OF NONSERIAL PRODUCTION-INSPECTION
     1SYSTEMS",/,5X,"BY GLOVERS SCHEME",///,5X,"** INPUT INFORMATION")
  112 FORMAT(//,5X,"** OPTIMAL SOLUTION")
   40 FORMAT(/,5X,"NO. OF OPS = ",I5,10X,"NO. OF SOURCES = ",I5)
   50 FORMAT(//,6X,"I",5X,"PA",6X,"PB",6X,"PE",6X,"CL",5X,"CM",
     16X,"CS",6X,"CV",//,(5X,I2,7F8.2))
   70 FORMAT(//,5X,"CK = ",F10.3,5X,"TOUT = ",F10.3,5X,"PAOQ = ",F10.3)
   90 FORMAT(/ ,5X,"MINIMUM COST = ",F15.6,//,7X,"I",4X,"D",8X,"PO",8X,
     1"PI",//,(5X,I3,2X,I3,2X,2F10.6))
  110 FORMAT(/ ,5X        ,"TOTAL NO. OF VECTORS ENUMERATED = ",I10)
  900 STOP
      END
```

```
      SUBROUTINE DSTMI(IA)
      DIMENSION IA(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOG,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),MR,IST,PINT
      K=MR-1
      DO 100 I=1,K
      IA(I)=1
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FFASI(IA)
      DIMENSION IA(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOG,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),MR,IST,PINT
      DO 100 I = 1, IN
      PO(I) = PE(I)
      PI(I) = PO(I) * (PA(I) + (1. - PA(I)) * IA(I))
  100 CONTINUE
      IJ=IN+1
      DO 200 J=IJ,NP
      TMULT = 1.
      JM=J-1
      DO 300 L=1,JM
      TMULT = TMULT * (1. - PI(L)) ** (RATE(L,J) / RATE(J,J))
  300 CONTINUE
      PO(J) = 1. - (1. - PE(J)) * TMULT
      PI(J) = PO(J) * (PA(J) + (1.-PA(J)) * IA(J))
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OBJCT(IA, ZD)
      DIMENSION IA(50)
      COMMON RATE(50,50), PA(50), PB(50), PF(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOG,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),MR,IST,PINT
      TSUM1 = 0.
      DO 100 I = 1, NP
      TSUM1=TSUM1+(RL(I)+RS(I)*(PB(I)+PO(I)*(1.-PA(I)-PB(I))))*
     1(1.-IA(I))
  100 CONTINUE
      ZD=TOUT*(TSUM1+CK*PI(NP))
      RETURN
      END



      SUBROUTINE ZTEST(IA, IB, Z)
      DIMENSION IA(50), IB(50), PONT(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOG,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),MR,IST,PINT
      TSUM1 = 0.
      TSUM2 = 0.
      DO 100 I = 1, NP
      PONT(I) = PO(I)
  100 CONTINUE
      PINT=PI(NP)
      CALL FEASI(IB)
      DO 200 J = 1, NP
      TSUM1=TSUM1+RL(J)+RS(J)*(PB(J)+PONT(J)+PONT(J)*(PA(J)+PB(J))*
     1IA(J))
      TSUM2=TSUM2+RL(J)*IB(J)+RS(J)*(PO(J)*(PA(J)+PB(J))+IB(J)*
     1(PO(J)+PB(J)))
  200 CONTINUE
      Z=TOUT*(TSUM1-TSUM2+CK*PINT)
      RETURN
      END



      SUBROUTINE EQUAL(IA, IB)
      DIMENSION IA(50), IB(50)
      COMMON RATE(50,50), PA(50), PB(50), PE(50), PO(50), PI(50),
     1CL(50), CM(50), CS(50), CK, NP, IN, TOUT,PAOG,RL(50),RS(50)
     2,CV(50),CE(50),PONT(50),MR,IST,PINT
      DO 100 I = 1, NP
      IB(I) = IA(I)
  100 CONTINUE
      RETURN
      END
```

Appendix D. A Sample Computer Solution Procedure.

```
/ØLD,Y10BH.
/LNH

10 2   1.          .01
   .05   .08   .08   .08 6.20   .09   .03   .07   .14 3.7
   .08   .01   .04   .21 2.60   .07   .09   .05   .26 3.8
   .01   .07   .05   .29 7.1    .02   .09   .08   .31 9.3
   .05   .01   .04   .39 6.40   .09   .01   .05   .43 8.3
   .03   .09   .07   .52 8.00   .08   .03   .02   .55 6.8
17   11          6
  2522                 3
     11  7  4
         7  2  5
            2     2
               5     5
                  2     2
                     5  3  2
                        1  1
                           1

/GET,GLØVER.
/GET,Y10BH.
/FTN,I=GLØVER,L=0.
     2.317 CP SECØNDS CØMPILATIØN TIME
/LGØ,Y10BH,BJY.
     3.746 CP SECØNDS EXECUTIØN TIME
/BANNER(BJY) SAMPLE RUN
 FILE BANNERED.
/DISPØSE,BJY=PR.
 YØUR ØUTPUT ID IS  T80223X
/LYE
```

APPENDIX E. Constants and Flow-Rate Matricies for Test Problems.

E.1. Constants

n=5

j	e_j	a_j	b_j	l_j	c_j
1	.09	.09	.01	.08	5.4
2	.07	.03	.01	.13	6.4
3	.07	.05	.07	.22	4.0
4	.05	.04	.03	.30	9.9
5	.02	.08	.03	.36	3.2

n=10

j	e_j	a_j	b_j	l_j	c_j
1	.05	.08	.08	.08	6.2
2	.09	.03	.07	.14	3.7
3	.08	.01	.04	.21	2.6
4	.07	.09	.05	.26	3.8
5	.01	.07	.05	.29	7.1
6	.02	.09	.08	.31	9.3
7	.05	.01	.04	.39	6.4
8	.09	.01	.05	.43	8.3
9	.03	.09	.07	.52	8.0
10	.08	.03	.02	.55	6.8

APPENDIX E.1 (Continued)

n=15

j	e_j	a_j	b_j	l_j	e_j
1	.08	.05	.01	.04	9.6
2	.05	.04	.07	.07	6.2
3	.06	.06	.01	.09	6.8
4	.03	.05	.04	.17	2.1
5	.03	.08	.01	.18	6.3
6	.09	.07	.08	.20	1.1
7	.02	.07	.01	.25	2.7
8	.07	.01	.06	.31	7.9
9	.06	.03	.05	.36	3.4
10	.08	.07	.05	.44	5.6
11	.01	.03	.08	.52	2.7
12	.03	.09	.08	.58	8.9
13	.05	.09	.05	.64	6.4
14	.03	.08	.05	.66	7.3
15	.03	.05	.06	.69	5.6

APPENDIX E.2. Flow-Rate Matricies

n=5

Structure A

$$R=\begin{bmatrix} 4 & 2 & 2 & 0 & 0 \\ 0 & 2 & 0 & 2 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Structure B

$$R=\begin{bmatrix} 10 & 4 & 5 & 1 & 0 \\ 0 & 2 & 0 & 0 & 2 \\ 0 & 0 & 5 & 2 & 3 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Structure C

$$R=\begin{bmatrix} 24 & 12 & 5 & 6 & 1 \\ 0 & 6 & 0 & 4 & 2 \\ 0 & 0 & 5 & 2 & 3 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

n=10

Structure A

$$R=\begin{bmatrix} 8 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 4 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

APPENDIX E.2. (Continued)

n=10

Structure B

$$R=\begin{bmatrix}
17 & 0 & 11 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\
0 & 25 & 22 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\
0 & 0 & 11 & 7 & 4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 7 & 2 & 5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 5 & 0 & 5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 3 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

Structure C

$$R=\begin{bmatrix}
70 & 0 & 48 & 0 & 12 & 0 & 10 & 0 & 0 & 0 \\
0 & 44 & 24 & 16 & 0 & 4 & 0 & 0 & 0 & 0 \\
0 & 0 & 24 & 0 & 24 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 16 & 0 & 8 & 0 & 8 & 0 & 0 \\
0 & 0 & 0 & 0 & 12 & 0 & 5 & 4 & 3 & 0 \\
0 & 0 & 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 2 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 2 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

n=15

Structure A

$$R=\begin{bmatrix}
3 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 0 & 6 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 8 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 3 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

APPENDIX E.2. (Continued)

n=15

Structure B

$$R=\begin{bmatrix}
13 & 0 & 0 & 4 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\
0 & 15 & 0 & 6 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 22 & 0 & 18 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 9 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 9 & 4 & 3 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

Structure C

$$R=\begin{bmatrix}
27 & 0 & 0 & 8 & 0 & 0 & 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\
0 & 52 & 0 & 16 & 28 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\
0 & 0 & 54 & 0 & 28 & 0 & 20 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 8 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 28 & 8 & 20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 8 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 20 & 8 & 8 & 4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 & 6 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 & 4 & 4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$