A COMPUTER-BASED MODEL FOR INSPECTION PLANNING

by

ROBERT RONALD TRIPPI

B.A., Queens College of the
City University of New York
(1966)

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF

SCIENCE

at the

MASSACHUSETTS INSTITUTE OF

TECHNOLOGY

June, 1968

Signature of Author ....
Alfred P. Sloan School of Management, May 17, 1968

Certified by ....
Thesis Supervisor

Accepted by ....
Chairman, Departmental Committee on Graduate Students

A COMPUTER-BASED MODEL

FOR INSPECTION PLANNING

by

Robert Ronald Trippi

Submitted to the Alfred P. Sloan School of Management on May 17, 1968
in partial fulfillment of the requirements for the degree of Master of
Science in Management

ABSTRACT

This thesis considers the problem of optimizing screening inspection effort in a general multistage sequential production process. Some of the factors relevant to obtaining optimal inspection policies are first described. A mathematical model is presented which possesses sufficient generality to produce interesting results, yet admits to a relatively simple solution. Thus, the potential for handling moderately large problems is present in this model. The mathematical model has been programmed on M.I.T.'s time-sharing system for the rapid solution and analysis of specific problems, and typical computational results are discussed. Through the analysis of the structure of the model and the resulting optimal soltuions, several significant insights into the determinants of the optimal placement of inspection points in the process are obtained, including the relative insensitivity of total quality cost to a suboptimal placement of inspection points and the expected behavior of the optimal policy as a result of changes in process parameters. The difficulties involved in treating more complex cases are also discussed, as well as possible extensions of the present model.

Thesis Advisor: Leon S. White

Title: Assistant Professor of Management

Professor E. Neal Hartley
Secretary of the Faculty
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Dear Professor Hartley:

In accordance with the requirements for graduation, I herewith submit a thesis entitled "A Computer-Based Model for Inspection Planning." I would like to take this opportunity to express my gratitude to Professor L.S. White for helping to provide me direction and motivation in the preparation of this thesis.

Sincerely,

Robert Ronald Trippi

TABLE OF CONTENTS

	Page
Chapter I - Introduction To The Problem	7
Organization	7
Some General Remarks Concerning Quality Assurance	7
Choice Of An Inspection Policy	9
Chapter II - The Mathematical Model	14
A Brief Review Of Relevant Literature	14
A Variation Of White's Shortest-Route Model	15
The Physical Problem	16
Network Formulations	18
Model I	19
Model II	21
Cost Structure	21
Network Solution	29
Flexibility Of The Model	31
Chapter III - Programmed Implementation	34
Computational Requirements	34
Programs, Data Files	35
Execution Sequence	39
Choice of Solution Structure	41
Computational Experience	41
Computational Limitations	44
Chapter IV - Experimental Results and Analysis	46
Data Set I - Example	46
Determinants Of The Solution Obtained	51

Table of Contents (Continued) | Page

Data Set II - Sensitivity of Solution to Processing Costs 56

Data Set III - Analysis of Fixed Inspection Costs 57

Chapter V - Conclusion And Recommendations For Future Study 61

Reflections on the Model 61

Promising Areas For Further Investigation 63

References 67

Appendix - Computer Programs 68

DIAGRAMS

Diagrams		Page
Figure 1	Sequential Production Process	17
Figure 2	Model I General Network Form	20
Figure 3a	Model II General Network Form	22
Figure 3b	Network For 2-Inspector, 3-Stage Line	23
Figure 4	File Structure.	38
Figure 5	Execution Sequence	40
Figure 6	Computation Time	43
Figure 7	Data Set I Total Cost Curve	48
Figure 8	TQC Vs. Fixed Inspection Cost	59

LISTS

List of Variables 33

Computational Results, Data Sets I and II 47

Computational Results, Data Set III 58

CHAPTER I

INTRODUCTION TO THE PROBLEM

Organization

This thesis is organized into five chapters and one appendix. This introductory chapter is an attempt to put the manufacturing inspection problem in proper perspective to the totality of problems in the shop and to suggest some of the considerations that one ought to be aware of in designing an inspection policy. In addition, a brief survey of relevant literature is presented.

Chapter Two is a description of the mathematical model which is the basis of the experimental work done by the author. Chapter Three explains the structure of the computational system and discusses computational limitations. Chapter Four describes the results of several problem runs with different data sets. Chapter Five is a commentary on the limitations of the model employed and suggests possibly significant related areas for future investigations.

The Appendix contains program listings and a brief description of each of the programs.

Some General Remarks Concerning Quality Assurance

Associated with virtually every production process are considerations concerning the quality of the product(s) outputted from the system. Even in those processes in which no apparent effort is expended in assuring a quality product, non-systematic, casual (perhaps visual) inspection is often an implicit, unavoidable part of the process. The present discussion will be limited to those manufacturing processes in which a systematic inspection procedure or policy can be devised in order to attain some goal (produce the product at minimum total cost, produce zero defects, etc.). It is interesting to note that under many problem formulations the quality aspects of the manufacturing system are embodied in both a goal statement (i.e. minimize total costs including the cost of assuring acceptable quality) and solution constraints (i.e. no more than X% defective finished products will be acceptable) simultaneously.

At the outset, we will assume inspection (of raw materials, partially finished goods, component sub-assemblies, and finished goods) is the primary instrument available for assuring acceptable quality in the finished product. Therefore, it is assumed that the technical production process to be employed is determined beforehand from considerations which are unaffected by a choice of inspection policy. Clearly, in the more general case, alternate manufacturing methods would affect both the frequency of defective operations occurring and the physical methods to be used in inspection.

The inspection policy chosen, as an integral part of the production process, will greatly affect other aspects of the system, such as facility scheduling, workforce requirements, etc. In particular, inspections which take place between manufacturing stages occupy finite time intervals and may thus be considered as operations in total facility scheduling. In this case the inspection policy must be determined before scheduling can take place.

Although optimal inspection policies may exist for flow- or job-shop situations in which items are manufactured in small or unit lots, it appears that the systematic determination of optimal inspection policies will be potentially most useful in those situations in which large lots of a product are produced at a time, due to the considerable collection of non-standard data and computational effort required, as will be seen shortly. The remainder of this thesis will be concerned with questions of "pure" inspection, ignoring possible interrelationships of inspection policy with other facets of production management, and taking the technological manufacturing process as given and not subject to change while the inspection policy is in effect.

Choice of an Inspection Policy

For the purposes of this paper the following definition of an inspection policy will suffice:

> An inspection policy is a statement of the defect types to be inspected for, the point within the production process at which inspection for each defect type is to take place, and the sampling processes to be employed.

We are not concerned here with the actual inspection or testing method used, whether it be mechanical, electrical, or visual. It is assumed that appropriate procedures can be devised by the engineering staff of the firm and that there are no choices to be made among alternate inspection procedures. What we seek is the allocation of inspection resources to each possible defect type at various points in the manufacturing process which will attain some predetermined goal. Sometimes an extreme solution, such as inspection of every operation immediately after execution or no inspection at all, will be optimal.

The factors affecting the choice of an inspection policy are of two general kinds: those associated with the manufacturing process (exclusive of inspection), and those associated primarily with inspection. Some factors associated with the manufacturing process which will in part determine the selection of an inspection policy are the following:

1) The arrangement of stages. In the simplest case manufacturing stages constitute a strictly-ordered sequence such that for any stage y there exists at most one stage such that x directly precedes y and there exists at most one stage z such that y directly precedes z. In more complex manufacturing situations assembly and partition operations may occur.

2) The defect-generating process. Defective operations may occur at manufacturing stages, thus imparting physical defects to the product. A defect generated at any one stage may be repairable at different costs, depending on its severity, or be non-repairable. Additionally, defects generated at a stage may be either dependent or independent of defects occurring at other (preceding) stages. In order to define the problem fully, statements must be made about the defect-generating process at each stage. These statements are usually of a statistical nature, specifying a probability distribution for each stage, or multivariate distributions for the case in which the defect-generating processes of several stages are dependent. For example, it may be observed that the defect-generating process at a particular stage can be modeled as Bernoulli with fixed parameter p, the probability of generating defect which is invariably non-repairable. The quantity 1-p would then be the probability of the operation being successfully executed.

The concept of a defect-generating process at each manufacturing stage can be broadened to include operations in which components are added to an assembly. It may be known, through incoming sampling procedures or otherwise, that a component (ex. a resistor) taken at random has a certain probability of being defective (out of spec). Thus, adding a defective component to an assembly can be considered to be a defect generated at the stage. A stage consisting of an assembly operation might then generate a component-type defect, an operation defect, or both.

3) Physical limitations on inspection imposed by the manufacturing process. In some instances, inspection for a defect generated at a manufacturing stage is impractical or impossible. For example, it may be a simple matter to test for a defect type within an assembly up until its outer casing is added, but impossible afterwards.

4) Processing costs. If a defect is discovered within an item which renders it unusable, any operations performed on the product after the defect occurred may be considered to be wasted and taken into account in determining an inspection policy.

5) Repair costs. Repairable defects may be repaired, usually at some cost.

6) Costs associated with the removal of worthless items or revenues gained from selling items no longer usable in the manufacturing process.

Some aspects of the inspection process relevant to the choice of inspection policy are:

1) The accuracy of inspection. The inspector may not be perfect. Defects of type i when inspected for may be overlooked with probability a_i. Also, there may be unavoidable ambiguities associated with inspection. For example, an electrical test of a partially completed assembly may indicate that one of several sub-assemblies is not functioning properly, but additional effort may be required to locate the defective component or wiring error within the correct sub-assembly.

2) Costs associated with inspection. Inspection costs may include labor, equipment, and utility costs. Some components of inspection cost may be fixed, others variable.

3) The availability of inspection resources. These are resources in limited supply, such as qualified manpower, special testing equipment, etc.

The above variables associated with the total production process which must be considered in selecting an inspection policy are meant only to be suggestive. Many other factors could no doubt be added to the list.

In addition to the factors just discussed, the effects of outputting defective goods must be considered in the selection of an optimal inspection policy. Defective finished goods may be returned to the factory for repair or exchange, usually at some cost to the firm. Customers may be lost temporarily or permanently, thus reducing future sales levels and profits. Certain legal restrictions on the quality of merchandise produced may apply.

In many instances it is company policy to maintain certain standards of quality although defects are unlikely to be observed by the consumer due to the nature of the product.

CHAPTER II

THE MATHEMATICAL MODEL

A Brief Review of Relevant Literature

The foundation of the most recent model-oriented papers is a result derived by Lindsay and Bishop (1964)[1] and White (1966)[2] regarding the intensity of inspection effort to be applied at those points in a single-line production process where inspection is to take place for the cases of non-repairable only and repairable only defect types, respectively. It has been shown that, under a fairly general cost structure including linear costs associated with outgoing defective material and per-unit inspection costs, a function including the total of inspection-related costs will be minimized by an extreme point solution at each stage, i.e. by zero or 100% inspection at each potential inspection point. This result will doubtlessly bring relief to many production managers, for 100% inspection at intermediate production stages appears to be common in industry. For models employing fixed costs associated with supporting an inspection station, one might expect this result to be further reinforced.

Pruzan and Jackson (1967)[3] have employed the "no partial sampling" theorem in the development of a model of inspection in a simple sequence of production stages, from which a least-expected-cost solution can be obtained through use of dynamic programming. Immediately following each manufacturing stage is a potential inspection point. From the set of potential inspection points a sub-set is chosen at which 100% inspection will take place. Each inspector then inspects for the defect types which may have occurred since the previous inspection point.

White (1967)[4] has developed a more general model similar to that of Pruzan and Jackson. Here, however, the defect-generating stages are partitioned into two disjoint sets: those which generate repairable defects and those which generate non-repairable defects. The optimization problem is cast into a shortest-route form, which admits to a relatively simple solution, and a formulation is given for constrained inspection resources.

It appears that future investigations of the inspection effort allocation problem are most needed in the areas of:

1) more complex manufacturing processes including the admissibility of assembly and partition stages, and
2) integration of optimal inspection policy search with the interdependent problems of facility scheduling, assembly-line balancing, work-force requirements, etc.

In addition, empirical evidence of the benefits to be gained through the use of formal analysis of this problem in an actual industrial environment would be welcomed.

A Variation of White's Shortest-Route Model

This chapter will be a detailed description of a model similar to that of White (1967), the major difference being that each manufacturing stage is considered to be a generator of both repairable and non-repairable defects. White's model considers each stage to be a generator of either repairable or non-repairable defects, but not both. We will henceforth refer to the property of being repairable or non-repairable as the "class" of the defect, while the stage of origin will identify the type of defect. The criterion to be used in the evaluation of inspection policies is the same as that in

White's original model - a minimal expected cost solution is sought.

In many manufacturing situations the most important division of defect types is into the repairable and non-repairable categories. It is this distinction between defects incurred at the same stage which will have the greatest influence on the manner in which the product is to be subsequently treated. For example, after a machining stage, items can often be reworked if too little material is removed during the operation, but may have to be scrapped if too much material is removed. Numerous similar situations easily come to mind. It is thus felt that a model which is to even approximately reflect reality should incorporate this feature in order to be applicable to a significant class of actual industrial settings.

<u>The Physical Problem</u>

We consider a production process consisting of an L-2 stage production line and stages 1 and L external to the line representing fictitious input and output activities. Potential inspection points exist after each production stage and will henceforth be identified with the manufacturing stage immediately preceding (Figure 1). Each stage j in the production line can generate type j repairable and type j non-repairable defects.

The following assumptions will be made:

1) A unit with at least one non-repairable defect will be considered a non-repairable item.

2) A unit with any number of repairable defects and with no non-repairable defects will be considered a repairable item.

3) After discovery of defects, repairable units are repaired and returned to the line at the point of inspection, and non-repairable units are removed from the manufacturing system.

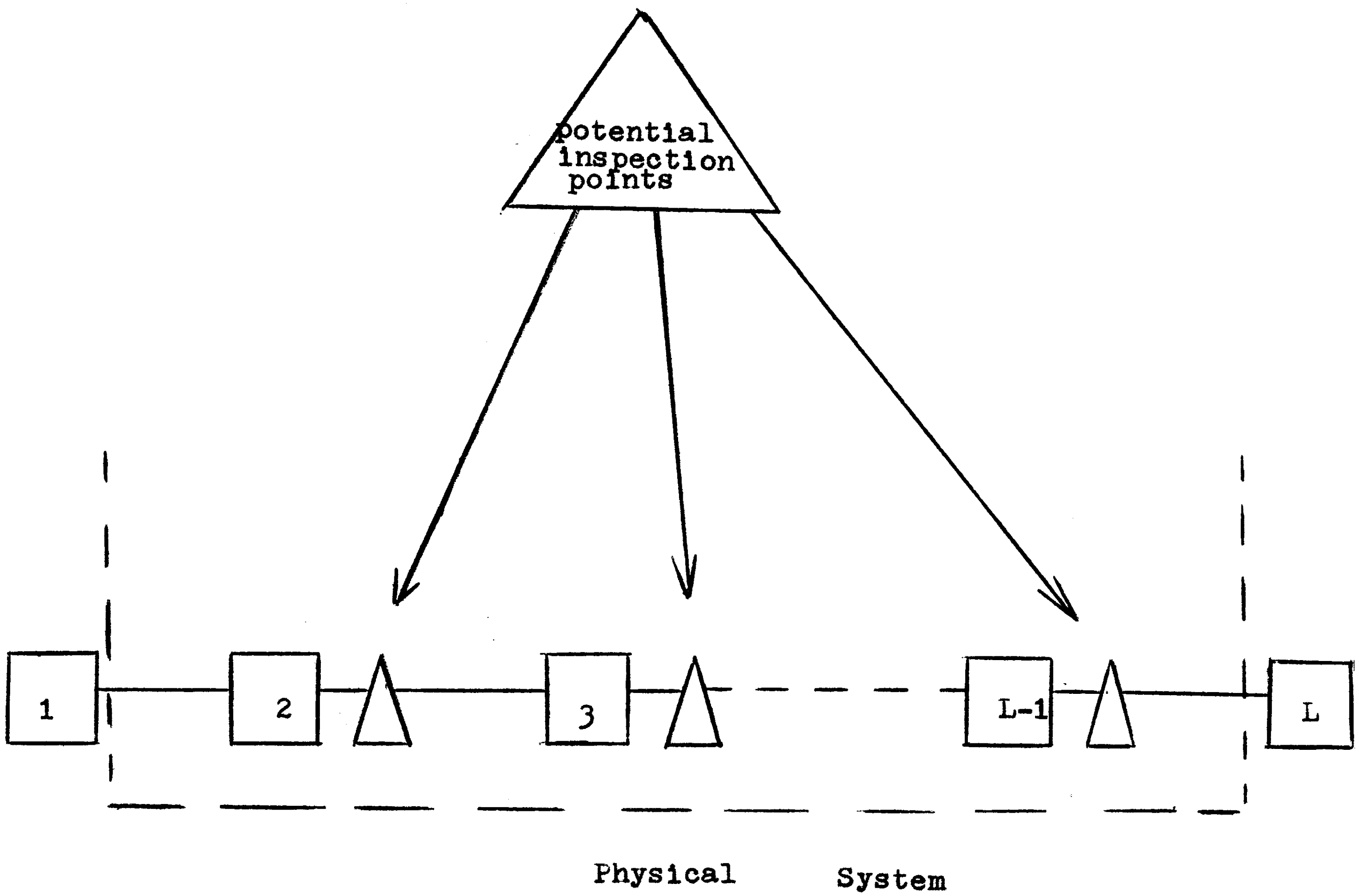


Figure 1

Sequential Production Process

4) Inspectors are perfect; defects are never overlooked.

5) Inspectors test for all defect types after the previous inspection point. An inspector at stage n thus inspects for defect types in the set (m+1, m+2, . .n), given that the previous inspection point is at stage m.

Initially a lot size, B_1, is assigned to the system, where B_j is taken to be the expected number of items leaving manufacturing stage j that are either perfect or repairable. Because of assumption (3), then, if there is an inspector assigned to stage k, there would be B_k items eventually leaving the stage to continue in the manufacturing process, and all units would be defect-free.

We assume that each stage generates defects independently of every other stage, and that the defect-generating process at each stage is multinomial with stationary parameters pr and pn. Thus, the mass function for e_j, the event occurring as an operation is performed on an item passing through stage j is:

$$f(e_{jk}) = \begin{cases} pr_j, & e_{j1} = \text{a repairable type-j defect is imparted to the item} \\ pn_j, & e_{j2} = \text{a non-repairable type-j defect is imparted to the item} \\ 1-pn_j-pr_j, & e_{j3} = \text{the operation is successfully executed} \end{cases}$$

An item leaving stage h which was last inspected at stage g may thus have any combination of properties e_{jk} for $g < j \leq h$, $k = 1,2,3$.

Network Formulations

At this point we will look ahead to the shortest-route model formulation in order to make clear the sort of information that will be needed in order to solve for an optimal inspection policy. We will compute the set of expected costs, c_{ij}, which represent the incremental cost incurred by having an inspection point at stage j of the manufacturing line given that the last inspection point is at stage i. Clearly, the cost of inspection at any stage j is a function of i. We thus seek a value for all c_{ij}, j=2,3...L, $i \leq j$. Let $\underline{\underline{C}} = \|c_{ij}\|$.

Once $\underline{\underline{C}}$ is known an optimal inspection policy can be obtained with the help of the Lindsay and Bishop theorem. Since any optimal inspection policy will specify either zero or 100% inspection at each potential inspection point, we must select from the set of all potential inspection points, K, the subset, $k \subset K$, at which items will be inspected 100%, and $k^c \subset K$ at which no inspection will take place, which will minimize the total expected cost. Equivalently, an inspection policy in this model can be defined as an L-2 component Boolean vector $(\delta_2, \delta_3, \ldots \delta_{L-1}) = \underline{\delta}$ in which $\delta_j = \begin{cases} 0 \text{ if no inspection occurs at stage } j \\ 1 \text{ if inspection occurs at } j \end{cases}$.

Model I

A shortest-route solution to the directed network shown in Figure 2 will give the minimal cost inspection policy desired if there are no limitations on the number of inspection points allowed. Here, c_{ij} represents the arc length from node i to node j. Upon solution for the shortest (lowest cost) route from node 1 to node L, δ_j=1 if node j lies on the shortest path; δ_j=0 otherwise. Nodes 1 and L serve a pedagogical purpose only; they provide a common origin and end for the network route.

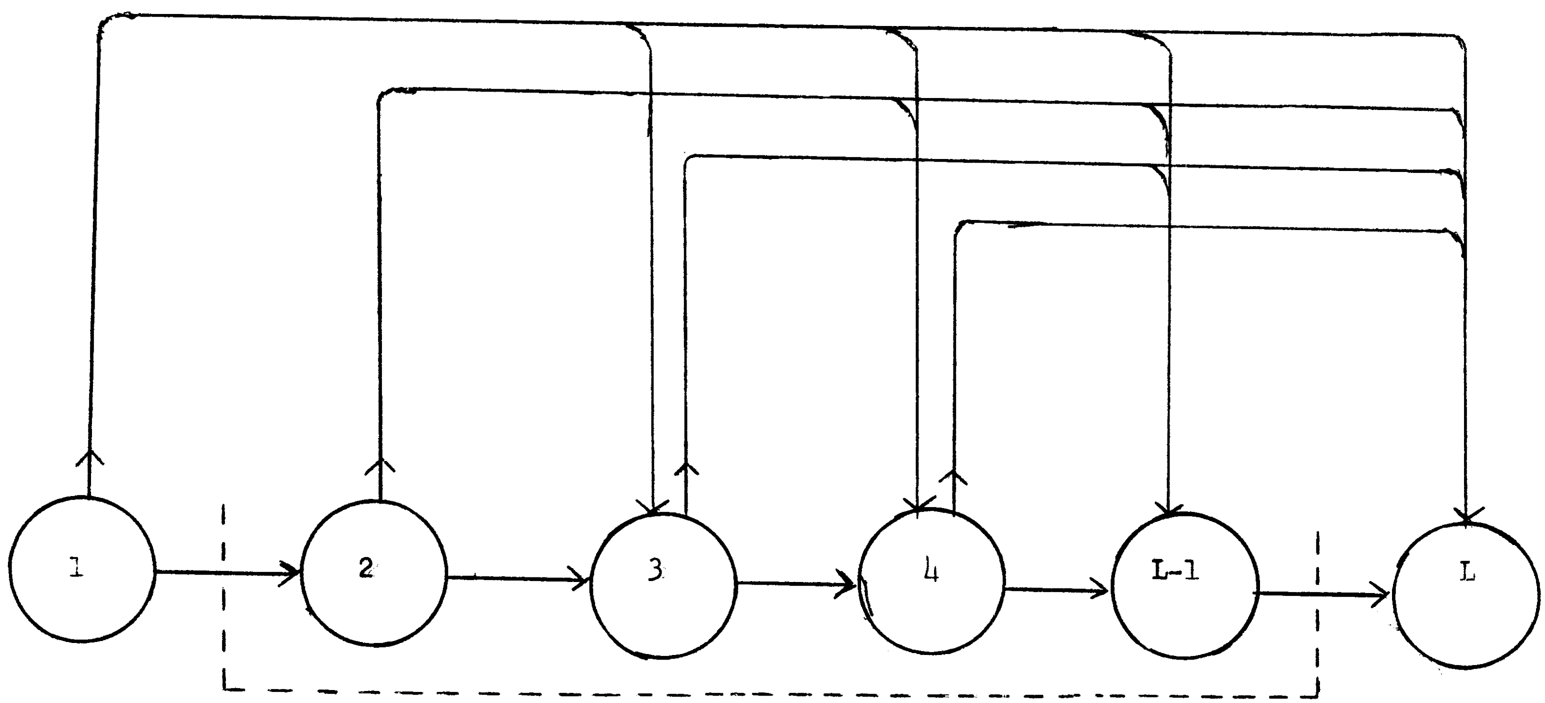


Figure 2 Model I General Network Form

Model II

If there is a limited number, n, of inspectors available for assignment to the L-2 potential inspection points, a multi-level shortest-route network of the form shown in Figures 3a and 3b can be solved for the optimal inspection policy. In this network there are n node "levels" plus the origin and end nodes, corresponding to the assignment of at most n inspectors to the potential inspection points. For those nodes on each level which lie on the shortest path from node (1,1) to node (n+2, L) $\delta_j^k = 1$, where j is the stage and k is the inspector number assigned to the stage plus one. $\delta_j^k = 0$ for all other nodes in the network and, necessarily δ_j^k must be zero for all $j < k$. If $\underline{\delta}^k$ is defined to be the L-2 component Boolean vector $(\delta_2^k, \delta_3^k \ldots\delta_{L-1}^k)$ for all levels k, then the optimal inspection policy is given by $\underline{\delta}^* = \Sigma_{k=2}^{n+1} \underline{\delta}^k$. In this network each arc (x,y), (z,w) is assigned cost c_{yw} from $\underline{\underline{C}}$.

Cost Structure

We will now formulate the classes of costs which comprise each c_{mn}, n=2..L, m=1..n-1.

A. Expected Cost of Scrappage:

$$ecs_{mn} = \sum_{j=m+2}^{n} cp_j(B_m-B_{j-1}) + cd_n(B_m-B_n) \qquad n=2..L-1,\ m=1..n-1.$$

This formulation is identical to White's. B_j has been defined previously to be the expected number of good or repairable items leaving stage j from an initial batch size of B_1. Thus, $B_j=B_1 \prod_{i=2}^{j} (1 - pn_i)$ j=2...L-1. The second component of ecs above represents the salvage value from disposal of non-repairable unfinished items, where cd_n is the market value (or cost of disposal) of a unit removed from the manufacturing process at stage n. Although

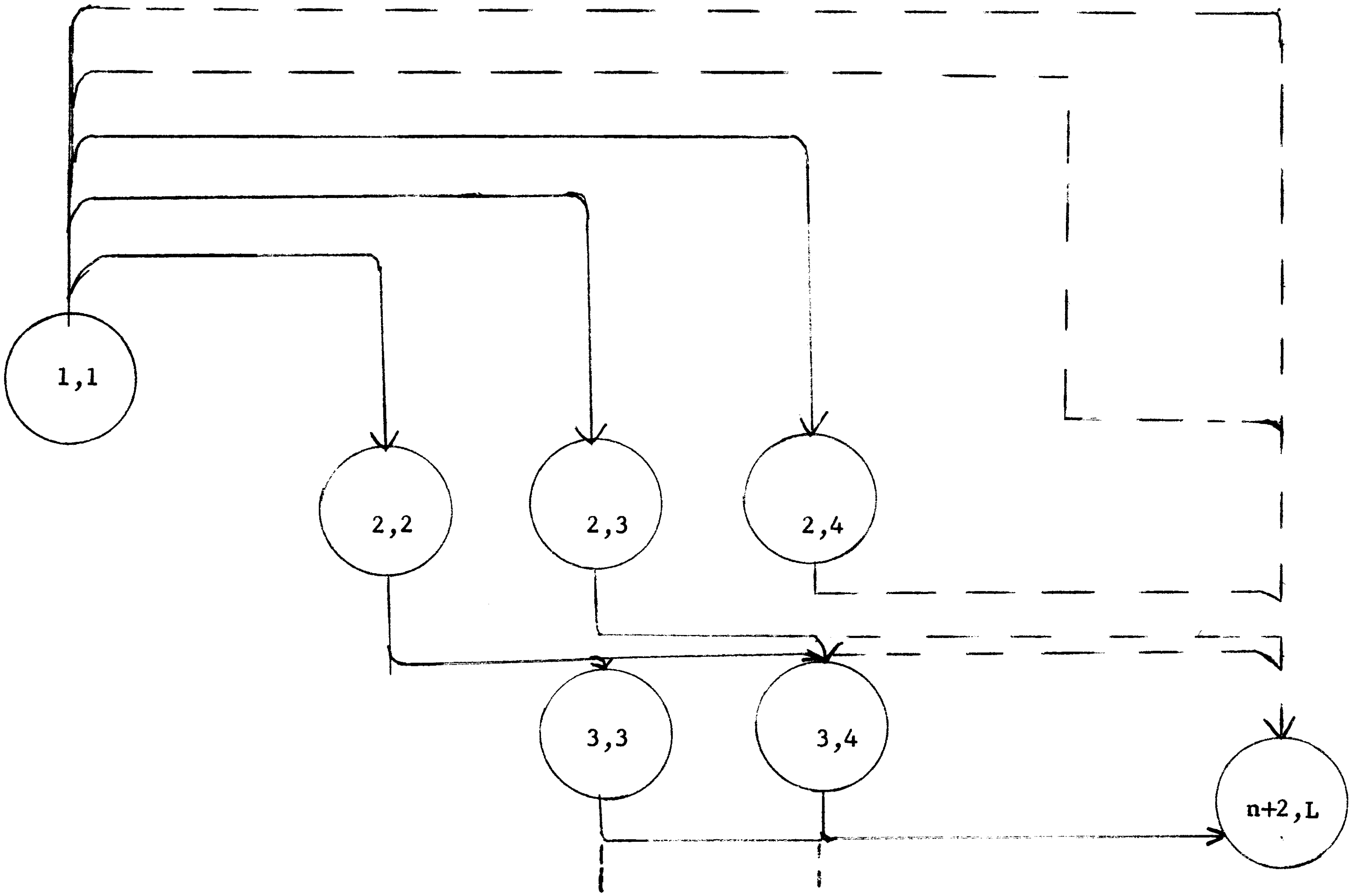


Figure 3a Model II General Network Form

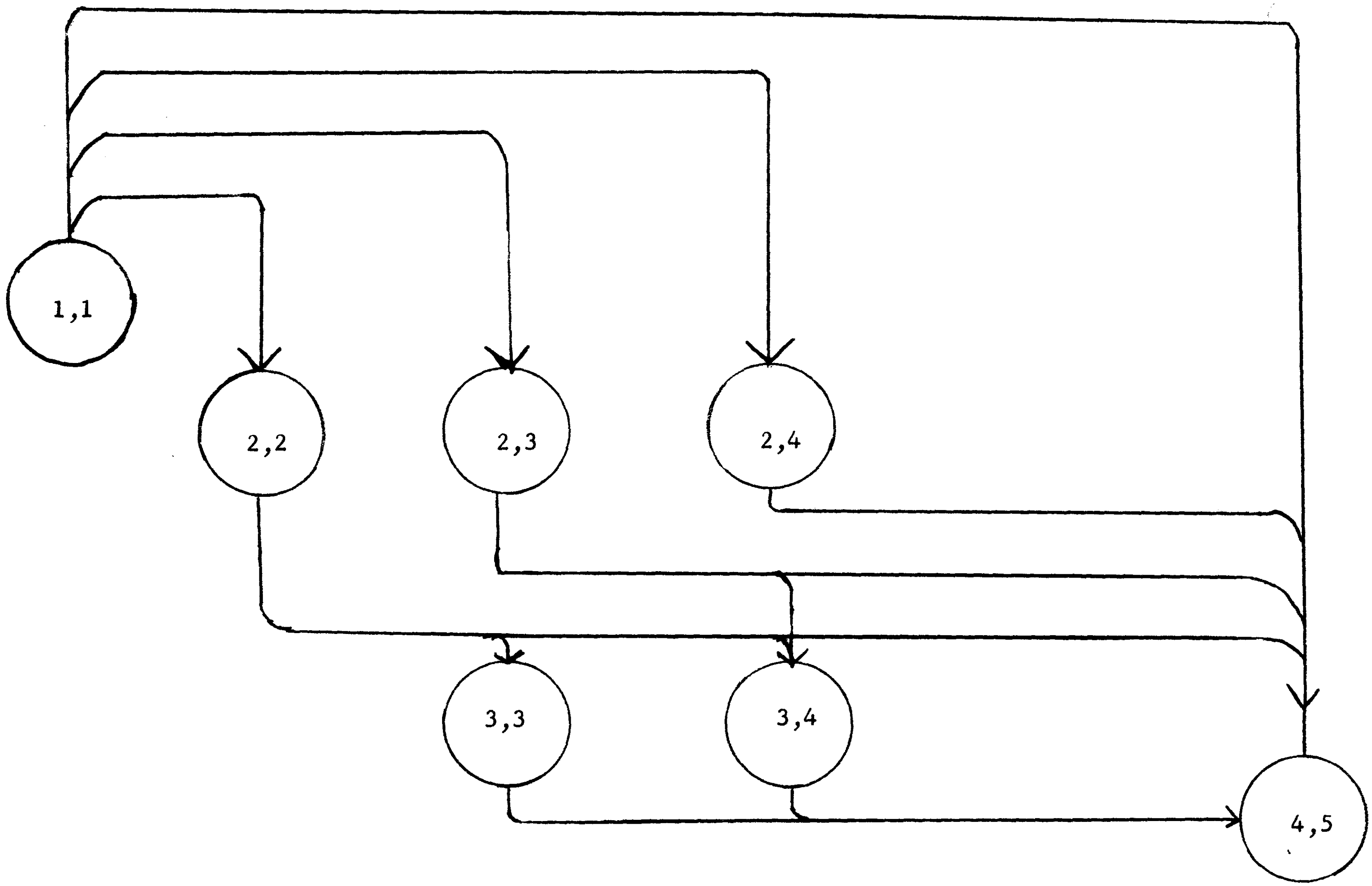


Figure 3b Network for 2-Inspector, 3-Stage Line

Number of Inspection Points Constrained to be Two or Fewer the value of cd might also depend on m, as suggested by White, the data collection difficulties would mitigate towards keeping this term as simple as possible.

The first component of the expected cost of scrappage above represents the wasted cost of processing items which already have acquired a non-repairable defect. With cp_j representing the cost of processing an item at stage j, the processing cost of units ruined at stage i is $(B_{i-1}-B_i) \sum_{j=i+1}^{n} cp_j$. Hence, the total expected processing loss between stages m and n is $\sum_{i=m+1}^{n-1} [(B_{i-1}-B_i) \sum_{j=i+1}^{n} cp_j]$ or $\sum_{i=m+1}^{n-1} \sum_{j=i+1}^{n} (B_{i-1}-B_i)\, cp_j$. Upon reversing the order of summation we obtain:

$$\sum_{j=m+2}^{n} \sum_{i=m+1}^{j-1} cp_j (B_{i-1}-B_i)$$

$$= \sum_{j=m+2}^{n} cp_j \sum_{i=m+1}^{j-1} (B_{i-1}-B_i)$$

$$= \sum_{j=m+2}^{n} cp_j \, (B_m-B_{m+1}+B_{m+1}-B_{m+2}+B_{m+2}\ldots+B_{j-2}-B_{j-1})$$

$$= \sum_{j=m+2}^{n} cp_j \, (B_m-B_{j-1})$$

B. Expected Repair Cost:

$$erc_{mn} = B_m [\sum_{i=m+1}^{n} cr_i pr_i] \prod_{j=m+1}^{n} (1-pn_j) \qquad n=2\ldots L-1,\ m=1..n-1,$$

where cr_i is the unit cost of repairing type i repairable defects.

The probability that there will be a repairable defect of type i and no non-repairable defects of type j, $m < j \leq n$ is:

$$pr_i \prod_{j=m+1}^{n} (1-pn_j).$$

Hence, the expected cost of repairing type i defects for B_m units is:

$$B_m\ cr_i\ pr_i \prod_{j=m+1}^{n} (1-pn_j)$$

Then the expected repair cost for all possible repairable defect types i between m and n is:

$$\sum_{i=m+1}^{n} (B_m\ cr_i\ pr_i \prod_{j=m+1}^{n} (1-pn_j))$$

$$= B_m(\sum_{i=m+1}^{n} cr_i\ pr_i) (\prod_{j=m+1}^{n} (1-pn_j))$$

Note that in general the cost of repair of a defect of type j is dependent also on the stage n at which it is discovered. Substituting cr_{in} for cr_i above does not affect the mechanics of calculation of erc in any way and is thus entirely feasible. In the working model to follow, however, the simpler form was chosen to make easier the task of data entry.

C. Expected Cost of Undetected Defects:

This is one of the most important categories of cost in the present model, the undesirable consequences of outputting defective products being the raison d'etre for quality assurance efforts. These costs may also be the most difficult to ascertain in any actual situation. Included may be the cost of handling and repairing returned items, the loss of company good-will, and the deterioration of dealer loyalty.

In this model two very simple formulations are given. The first assumes that the cost of undetected defects of different types are additive. In this case the expected cost of undetected defects is

$$ecu1_{mL} = B_m \sum_{j=m+1}^{L-1} (pn_j + pr_j) \; cul_j \qquad m = 1...L-2$$

$$= 0 \qquad m = L-1$$

where cul_j is the cost associated with a type j defect leaving the plant.

If we assume a fixed penalty cost for a unit with any defect type or combination of defects leaving the plant, then

$$ecu2_{mL} = B_m \, cu2 \, [1 - \prod_{j=m+1}^{L-1} (1 - (pr_j + pn_j))] \qquad m = 1..L-2$$

$$= 0 \qquad m = L-1$$

where cu2 is the penalty cost for outputting a defective item. Either of these two expressions will give us the arc cost from any node, m, to the dummy end node, L, signifying that inspection last takes place at stage m.

D. Expected Inspection Cost:

White's model contains expressions for the fixed cost of an inspection station plus the per-unit inspection cost at the station. However, it is not at all clear how these costs may be derived. One strategy is to assume that the expected variable inspection cost from inspecting at stage n given that inspection last took place at stage m is a function of both the set of defect types to be tested or inspected for the efficiency of inspection as determined by the order in which inspection takes place. Knowledge of m and n alone uniquely determines the set, S, of defect types to be inspected over at stage n. Thus, $S = (j \ / \ m < j \leq n)$. We now seek a least-expected-cost inspection sequence over S.

The assumptions about the inspection procedure to be used can be summarized as follows:

1. There is a unit cost, C_j, associated with each defect type that is tested for.
2. The inspector inspects or tests for defects in the set S according to some predetermined least-expected-cost sequence Q^* for all items.
3. As soon as the first non-repairable defect of any type is discovered (if any), inspection ceases for that item, it is put aside, and the inspection sequence begins anew for the next item.

Thus, the unit expected inspection cost, uic, for any sequential ordering, Q, of the elements in S is:

$$uic(Q) = C_1 + (1-pn_1)C_2 + (1-pn_1)(1-pn_2)C_3 + \dots + (1-pn_1)(1-pn_2)\dots(1-pn_{k-1})C_k$$

Where there are k elements in S and

C_i = the unit inspection cost for the defect type inspected for in sequential position i of Q.

We seek the permutation, Q^*, of the elements of S which, when used as a sequential inspection order, will minimize uic_{mn}.

Price[5] has identified this problem and offered the suggestion that an optimal sequence can be obtained through complete combinatorial enumeration. Thus, if there are k elements in S, k! orderings must be generated, and uic(Q) computed for each in order to identify the optimal sequence. This is clearly a computationally undesirable approach to solution of a problem which arises $\frac{(L-2)(L-1)}{2}$ times in the inspection allocation model. Derman[6] has investigated this problem and discovered a simple rule which will give the optimal inspection sequence for an inspection sequencing problem of which this is a special case. Johnson[7] has considered an inspection situation under rather different inspection and repair assumptions, but the logical arguments used are equally applicable in this case.

<u>Theorem</u>:

Number the k defect types in S according to increasing value of $\frac{C_j}{pn_j}$. This is the optimal order of inspection.

<u>Proof</u>:

Let Q' be the sequence Q after interchanging components i and i+1 where $i+1 \leq k$. Then we have:

$$uic(Q')-uic(Q) = \left(\prod_{j=1}^{i-1} (1-pn_j)\right) \left((C_{i+1}+(1-pn_{i+1})C_i)-(C_i+(1-pn_i)C_{i+1})\right)$$

$$= Y(pn_i C_{i+1} - pn_{i+1} C_i)$$

which is $\begin{matrix}\text{positive}\\ 0\\ \text{negative}\end{matrix}$ according as $\frac{C_i}{pn_i} \begin{matrix}<\\=\\>\end{matrix} \frac{C_{i+1}}{pn_{i+1}}$,

a transitive relation. If we successively interchange consecutive components wherever this difference is negative, we are thus led to the rule that inspection for defect type i precedes j if

$$\frac{C_i}{pn_i} < \frac{C_j}{pn_j} \quad .$$

If the optimal inspection sequence is Q*, the total inspection cost associated with inspection at n given that inspection last takes place at m is:

$$eic_{mn} = B_m \; uic(Q^*)_{mn} + fic_n, \; n=2...L-1, \; m=1..n-1$$

where fic_n is the cost of locating an inspection station at stage n independent of the testing actually done (i.e. a fixed cost).

As an aside, it should be fairly evident that for potential inspection points at any numbered stages a < b < ..z,

$$uic^*_{az} \leq uic^*_{ab} + uic^*_{bc} + \ldots\ldots + uic^*_{yz}.$$

The cost matrix $\underline{\underline{C}}$ can be derived from the above cost classes as follows:

$$c_{mn} = eic_{mn} + erc_{mn} + ecs_{mn} \qquad n=2...L-1, \; m=1...n-1$$

$$c_{mL} = ecu1_{mL} \text{ or } ecu2_{mL} \qquad m=1...L-1.$$

<u>Network Solution</u>

It is well-known that the directed shortest-route problem can be visualized as a transhipment problem in which there is an excess of one unit at the source node, a deficit of one unit at the sink node, and intermediate nodes have neither deficit nor excess units. The object is thus to transport the unit from source to sink at minimum cost. The mathematical problem is:

$$\text{minimize } x_o = \sum_{i=1}^{m-1} \sum_{j\varepsilon R_i} c_{ij}\, x_{ij}$$

s.t.

$$\sum_{j\varepsilon R_1} x_{ij} = 1$$

$$\sum_{i\varepsilon S_k} x_{ik} - \sum_{j\varepsilon R_k} x_{kj} = o \qquad k = 2,3\ldots m-1$$

$$\sum_{i\varepsilon S_m} -x_{im} = -1$$

$$x_{ij} = x_{ij}^{\;2} \qquad \forall\ i,j$$

Since x_{ij} represents the "quantity shipped" from node i to node j, $x_{ij}=1$ indicates that path (i-j) lies on the lowest-cost route, while $x_{ij}=0$ indicates that path (i-j) is not a part of the lowest-cost route. R_k represents the set of nodes immediately following node k, while S_k represents the set of nodes immediately preceding node k. For example, in Model I $R_k=(j\ /\ j>k)$, $S_k = (i\ /\ i<k)$, and m=L.

The dual to the transhipment problem above is:

$$\text{maximize } y_o = y_1 - y_m$$

$$\text{s.t. } y_j - y_i \leq c_{ij} \qquad \forall i,\ \forall\ j\varepsilon V_i$$

y unconstrained in sign

In Model I V_i is thus $(j\ /\ j>i)$. Y_1 is arbitrarily set to 0.

Ford[8] has devised an extremely fast algorithm for solving the dual problem above, exploiting the fact that each row contains but two variables:

> ..Assign initially $y_o = 0$ and $y_i = \infty$ for $i \neq 0$. Scan the network for a pair i and j with the property that $y_i - y_j > c_{ji}$. For this pair replace y_i by $y_j + C_{ji}$. Continue this process. Eventually no such pairs can be found, and y_m is now minimal and represents the minimal distance from 0 to m...

A dynamic programming approach to the problem above is similar, requiring only an ordering of the rows in Ford's algorithm so that only one scan of the inequality set is necessary.

From the Duality Theorem we know that $x_o^* = y_o^* = -y_m^*$. Thus, a solution to the maximization problem above will give the value of the minimum-cost allocation of inspectors in the inspection model. From the complementary slackness properties of the primal and dual problems (see Dantzig[9]) it follows that for each row in the dual that is an equality, $y_j^* - y_i^* = c_{ij}$, the corresponding primal variable, x_{ij}^*, is ≥ 0. x_{ij}^* must equal 0 for all other i,j pairs. If we set each of the x_{ij}'s in the first group equal to 1, a spanning tree for the network will result, indicating the least-cost route from the origin node to every node in the network. Note that this is not a solution to the primal problem, since we will have included more paths than are necessary to traverse the network from origin to terminal nodes. However, the unbroken route from the initial to the terminal node can now be easily identified, the associated variables of which comprise a solution to the primal problem.

Flexibility of the Model

The model presented is quite flexible in its ability to cope with multiple kinds of defects occurring at each stage. The partition of defects into repairable and non-repairable classes has already been discussed. In an actual industrial situation, however, one stage of a manufacturing process (one operation) may generate more than one kind of repairable or non-repairable defect. For example, adding a component to an assembly may result in a faulty mechanical connection or a poor electrical connection or both. Different costs of repair may arise from these two defect kinds. In the case in which variable inspection cost for defect type j is constant regardless of the number of kinds (both repairable and non-repairable) of type j defects which may occur, the analysis is straightforward. For k repairable type j defects possible and 1 non-repairable type j defects possible let

$$pn_j = 1 - \prod_{i=1}^{1} (1-pn_{ji})$$

where pn_{ji} is the probability of an item acquiring a type j repairable defect. Similarly, let

$$pr_j = 1 - \prod_{i=1}^{k} (1-pr_{ji}).$$

Consequently, we may take as the repair cost of type j repairable defects (which are now of several kinds) the expected cost of type j defects over the k kinds:

$$cr_j = \frac{\sum_{i=1}^{k} cr_{ji}\, pr_{ji}}{\sum_{i=1}^{k} pr_{ji}}$$

where cr_{ji} is the cost of repairing type j repairable defects of kind or severity i. Note that under the assumptions of the model multiple defect kinds affect only pn_j, pr_j, and cr_j. All cost functions are then based on the values of these variables obtained as above.

List of Variables

B_j = the expected number of good or repairable items leaving stage j.

C_j = per-unit variable inspection cost for type j defects.

$\underline{\underline{C}}$ = matrix of cost coefficients $//c_{mn}//$.

c_{mn} = incremental cost of having inspection at stage n given that inspection last takes place at stage m = $\mathbf{d}_{mn}$ in programs.

cd_j = disposal cost (or salvage value) of defective items removed at stage j.

cp_j = unit processing cost at stage j.

cr_j = unit repair cost of type j repairable defect.

$cu1_j$ = cost of a type j defect outputted.

cu2 = cost of a defective unit outputted.

ecs_{mn} = expected cost of scrappage component of c_{mn}.

eic_{mn} = expected cost of inspection component of c_{mn}.

erc_{mn} = expected repair cost component of c_{mn}.

$ecu1_{mL}$ = expected cost of undetected defects assuming additive costs compon-end of c_{mn}.

$ecu2_{mL}$ = expected cost of undetected defects assuming constant costs component of c_{mn}.

fic_j = fixed cost of inspection at stage j.

L = # of stages in model = # of actual production stages plus two.

NSPECT = maximum # of inspection locations (in SHORT2).

pn_j = probability of a unit acquiring a type j non-repairable defect.

pr_j = probability of a unit acquiring a type j repairable defect.

CHAPTER III

PROGRAMMED IMPLEMENTATION

Computational Requirements

A programmed computational system for obtaining an optimum solution to the inspection problem modeled in Chapter Two is to be described here. A set of ten free-standing computer programs perform the necessary calculations. The basic requirements for this program set are as follows:

1) Efficient computation of arc costs and evaluation of networks for solutions to unconstrained, constrained, and arbitrary policy cost problems. It is desired further to be able to accomodate problems involving a large enough number of stages to discern patterns in inspection station placement in subsequent data runs.

2) Provision of problem solutions with a maximum of flexibility. For example, a convenient method for entering data vectors is desired. In addition, a minimum of recalculation should be required for changes in the data set, within the limitations of programming effort available.

3) Need for a minimum of human solution effort once data has been put into machine readable form. Since the evaluation of the unconstrained, constrained, and arbitrary policy problems will usually be required of each data set, task initiation should be as simple as possible.

Programs, Data Files

The ten calculation routines are free-standing programs written in the Fortran IV language and are stored in the user's allotted filestorage area on the main disc of the M.I.T. CTTS time-sharing computer facility (Computation Center). Time-sharing provides much of the flexibility and ease of human intervention desired of this problem-solving system. The user, communicating on-line with the IBM 7094 computer via a typewriter-like console, can enter, compile, and select programs for execution, as well as establish data storage files. Cost and parameter data are initially entered on pseudo-tapes (actually one or more disc records) via the console in the form of strings of numbers. The file structure employed is illustrated in Figure 4. CTTS allows the user to load and run object programs utilizing either pre-stored data or data entered from the console at execution times. Each program in this set, as it proceeds, reads the data it requires from the pseudo-tape files and assigns these values to the appropriate variables. In some of the programs presented herein, requests for additional information are printed on the console to elicit the user's reply. The programs are:

1) MASTER. This program will calculate B_m, m=1...L-1, the expected number of good or repairable units leaving stage m. An assumed initial batch size of 1000 units provides good scaling for all of the calculations.

2) EIC. Expected inspection cost, eic_{mn}, is calculated for all feasible arcs.

3) ERC. Expected repair cost, erc_{mn}, is calculated for all feasible arcs.

4) ECS. Expected cost of scrappage, ecs_{mn}, is calculated for all feasible arcs.

5) ECU1. Expected cost of undetected defects under the additive cost assumption, $ecu1_{mL}$, is calculated for m=1..L-1.

6) ECU2. Expected cost of undetected defects under the constant cost assumption, $ecu2_{mL}$, is calculated for m=1...L-1.

7) AGGREG. This program will aggregate the components of arc costs, eic, erc, ecs, and either ecu1 or ecu2 to yield the arc cost matrix $\underline{\underline{C}}$. A message is printed on the console asking the user whether he wants to employ additive or constant costs of undetected defects. The user's response selects either ecu1 or ecu2 to provide c_{mL}, m=1..L-1.

8) SHORT. This program will evaluate the shortest-route network corresponding to the unconstrained inspection problem, using as input the values of $\underline{\underline{C}}$ stored on an intermediate data pseudo-tape. This program will print the minimum total cost of the optimal policy obtained and the locations at which inspection stations should be placed. In addition, the minimum arc costs from node 1 to all other nodes (the spanning tree) are listed for further analysis.

9) SHORT2. This program will find the optimal inspection policy for the constrained inspection problem. The user is asked to enter on the console only the maximum number of inspection stations to be allowed. All other work, including building up the extended network of feasible arcs and assigning arc costs, is done by the program. The console will print the optimal location of inspection stations and the total quality cost of the policy.

10) LONG. This program will evaluate the total quality cost of any arbitrary inspection policy. The user is asked only to enter on the console the numbers of the stages at which inspection is to take place. The program will then print on the console the total cost of the policy entered.

Each program above will read in the required data from the appropriate pseudo-tapes and output data onto the appropriate intermediate files as shown in Figure 4. More detailed descriptions of these programs and statement listings may be found in the appendix.

FIGURE 4

FILE STRUCTURE

Input and intermediate variables stored in pseudo-tapes (one or more disc records)

.tape.	variables (input)
2	L
4	$pn_2 \ldots pn_{L-1}$
5	$pr_2 \ldots pr_{L-1}$
6	$C_2 \ldots C_{L-1}$
7	$cp_2 \ldots cp_{L-1}$
8	$cd_2 \ldots cd_{L-1}$
9	$cr_2 \ldots cr_{L-1}$
10	$cu1_2 \ldots cu1_{L-1}$
11	cu2
12	$fic_2 \ldots fic_{L-1}$

.tape.	variables (intermediate)	
20	$B_1 \ldots B_{L-1}$	
21	ϕ_{mn}..m=1..L-1..n=m+1...L	(m is major index)
22	$eic_{m,n}$ m=1..L-2, n=m+1.L-1	(m is major index)
23	erc_{mn} m=1..L-2, n=m+1..L-1	(m is major index)
25	ecs_{mn} m=1...L-2, n=m+1...L-1	(m is major index)
24	$ecu1_{mL}$ m=1...L-1	
26	$ecu2_{mL}$ m=1...L-1	

Execution Sequence

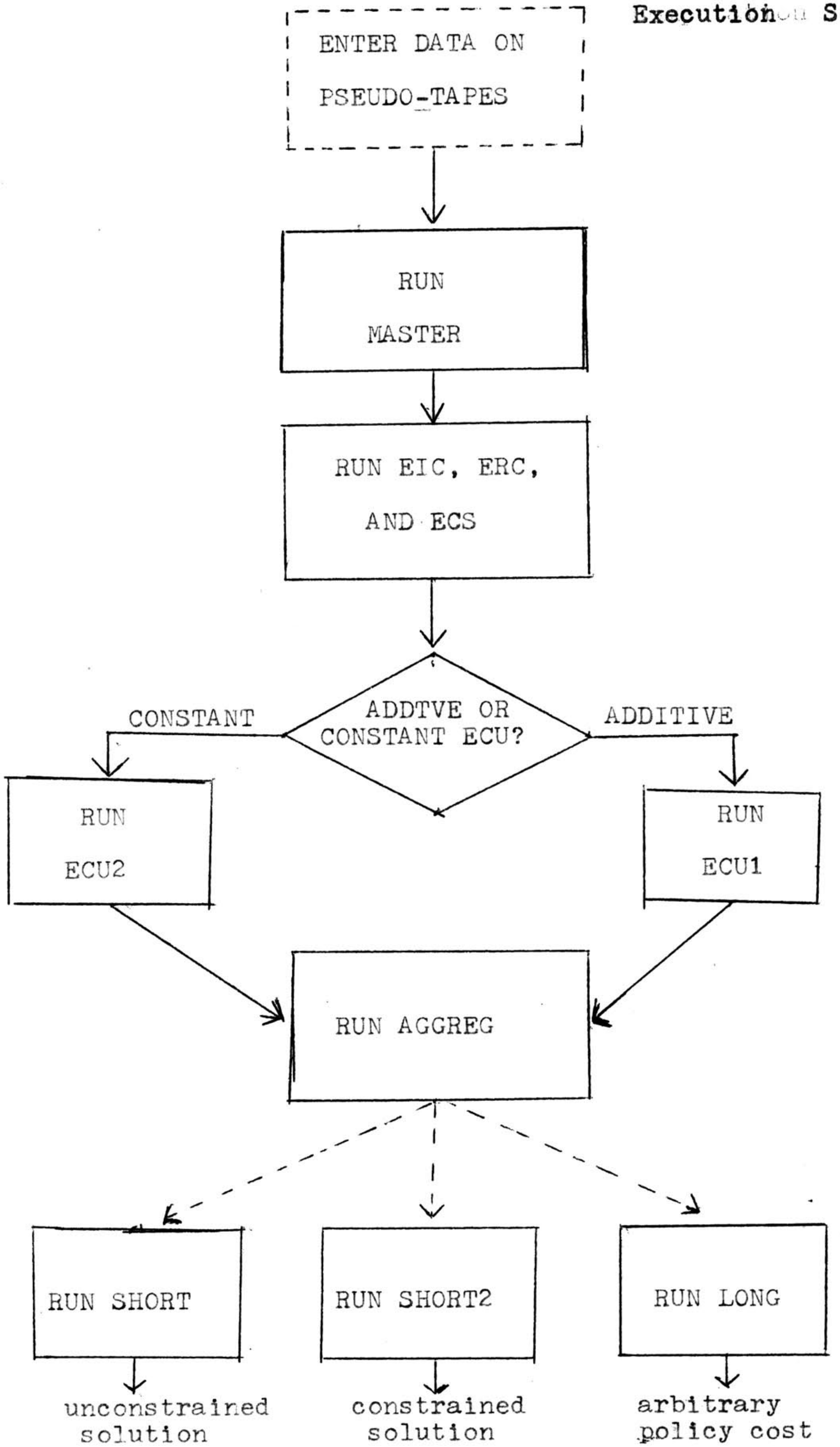
Input data must be entered in the proper pseudo-tape files before any calculation can take place. Each object program may be loaded and executed under CTTS through the use of one or two simple commands communicated via the console.

Starting with new data, the complete solution requires the execution of several programs in sequence, as illustrated in the flow chart of Figure 5. First, MASTER must be executed to provide the vector B which is utilized as input to the arc cost component programs. Next, EIC, ERC, and ECS are executed in any order to provide arc-cost components. ECU1 or ECU2 or both must also be executed at this time to provide c_{mL}, m=1..L-1, although only one of these data sets will be used in aggregation. AGGREG is next executed to aggregate erc, eic, and ecs, providing arc costs, and either ecul or ecu2 provides terminal arc costs. The program requests the user to indicate on the console whether he desires the additive or constant cost formulation of expected undetected defect costs.

Once the arc-length matric, $\underline{\underline{C}}$, has been outputted, one or more of the network routines, SHORT, SHORT2, or LONG, may be executed to yield policies and total quality costs. In addition, if only some data entries on pseudo-tapes are revised, only the programs to do the affected calculations need be re-executed. Only if the pr or pn vectors are changed must the full sequence of programs be rerun, as all arc cost components include elements of the vector B.

figure 5

Execution Sequence



Choice of Solution Structure

Separate programs for arc-cost calculation were adopted to permit re-execution of fewer than all of the programs if data elements of one or two types only are altered. This has already been mentioned. The use of a control program which would execute several or all of the programs described above as subroutines was considered, but this idea was discarded since it would require at least as much user effort in task specification as is presently required in executing the free-standing programs in sequence. The use of pseudo-tape files for storage of input data permits data entry via the console prior to program execution at the user's convenience and rapid read-in of data from files (disc records) to core as each program is executed. As mentioned earlier, the input data sets are limited to vectors for ease of manual data entry, although some data element types may be considered to be matrices in the general mathematical model (unit repair costs, for example).

Storage of the elements of the arc-cost matrix $\underline{\underline{C}}$ on intermediate pseudo-tapes permits the evaluation of constrained optimal, unconstrained optimal, and arbitrary policies utilizing the same set of final arc-cost data. Thus, the effects of constrained inspection resources of various degrees and existing policies can be readily compared without additional arithmetic calculations.

Computational Experience

There appear to be no major computational difficulties associated with the inspection model. Calculation of arc costs is straightforward as described, although perhaps unsparing of computer time for problems of greater size than those considered here. This remains to be seen. The following total computation times are typical for lines of the length indicated.

L = # of stages (including initial and terminal)	Total Time
5	47.93 sec
12	52.28
22	63.05
50	134.13

These data are plotted in Figure 6. We cannot, however, place much emphasis on the reproductibility of these figures since (1) total time includes on the average 20% swap time, which may vary from run to run with the same data and program, and (2) total time includes program and data file retrival and program load time of a setup nature.

Also, since the number of calculations required for the 5-stage (3 physical stages) problem is quite small we may assume that virtually all of the 47+ seconds required for solution to this problem is of a setup nature and represents a fixed cost of using separate programs and the file search time of the time-sharing system.

It is interesting to note, however, that the number of arcs for which cost components must be calculated is approximately equal to $\frac{s^2}{2}$, where s is the number of stages in the system, and that the average arc-cost component calculation is roughly proportional to the number of stages that the arc spans. We might then expect a priori that an approximately cubic relationship exists between arc-cost calculation times and the number of production stages. Moreover, approximately $\frac{s^2}{2}$ calculations are required for the unconstrained network solution, MASTER, and AGGREG. Therefore, total problem solution times should increase initially somewhat less rapidly than the cube

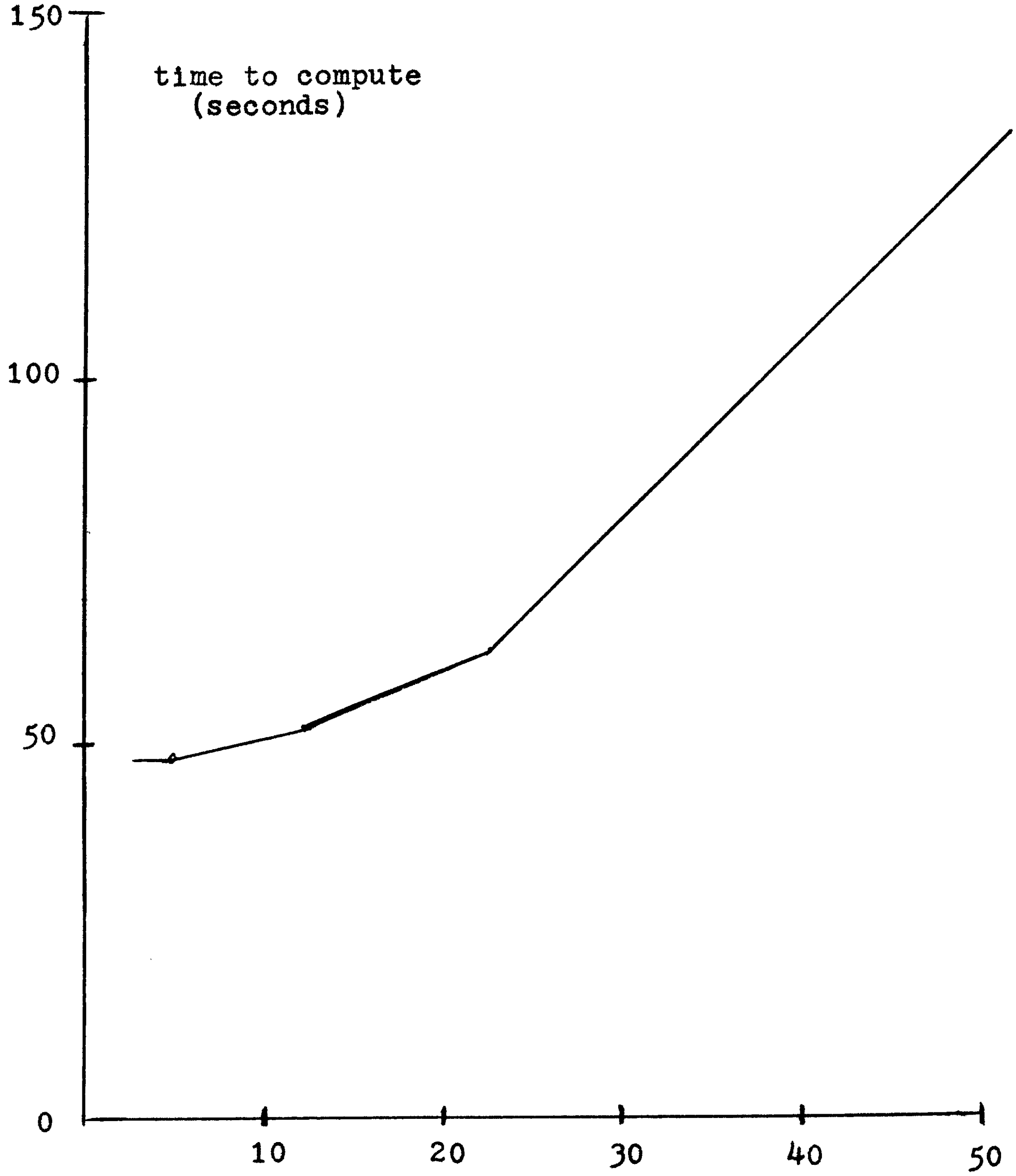


Figure 6

Computation Time

of the number of stages, but more rapidly than the square. This is suggested also by the curve on Figure 6 which increases less rapidly than a function of a cubic term only, but more rapidly than that of a square term only. The total time function might then be of the form $c+as^2+bs^3$. For lines of many stages we would expect the cubic term to dominate, rapidly limiting the maximum problem size that can be economically handled. Further empirical investigations with larger problems would be necessary in order to validate the hypothesized computation time relation. These were not attempted in the present investigation due primarily to the inability of SHORT2 to build up feasible arc identification vectors for problems significantly larger than the ones tested without approaching the 32k user available core capacity of CTSS.

Computational Limitations

The program set employed in this investigation is designed to allow maximum flexibility for experimentation and evaluation of parameter structure on inspection policies. No effort has been made either to minimize computation times or to solve problems of the largest size. It is likely that the elimination of separate free-standing programs for different solution steps would significantly reduce the program find-and-load time of 47+ seconds. Batch processing with a more powerful machine than the 7094 and more efficient programming might also significantly reduce solution time. However, the cubic relationship between line length and total solution time must still be reckoned with.

In an implementation designed specifically to handle large problems arc-cost components might be aggregated as calculated, to eliminate the loading of three matrices and one vector into core to be aggregated into the matrix of arc costs, $\underline{\underline{C}}$. Thus matrices with three times as many elements may be handled with the same core capacity as with the present set-up. A machine with a large fast core storage area available to the user (on the order of 150k data words) should, with proper programming, be able to provide at least optimal unconstrained solutions for problems of about 300 stages if the entire arc cost matrix is loaded into core prior to network calculations. This is not at all a requirement, however, since the unconstrained and constrained network algorithms can utilize parts of this cost matrix at a time to evaluate minimum-cost routes from the initial node to all other nodes working sequentially from lower-order nodes to higher-order ones, saving only the minimum cost to each node already evaluated. Thus, in theory and with relatively little change in the SHORT and SHORT2 programs the solution to problems of much larger size than 300 stages is potentially feasible if, in addition, arc-length component costs are removed to supplementary storage periodically as they are calculated by the arc-cost routine and accumulate.

Perhaps more significant than the feasibility and economics of calculation for large problems is the data-gathering effort required. It is, for example, unlikely that the probabilities associated with the defect-generating processes at each stage will be immediately available, unless records of defect repair have been kept in the past. A study may thus have to be undertaken, in which there would in many instances be a high likelihood of data contamination resulting from attention focused on the production line. In addition, cost components such as repair costs and penalty costs for undetected defects may be non-standard or unmeasurable. Thus, collection of both the quantity and the types of data necessary in order to implement this model may prove to be a most challenging task in an actual industrial environment.

CHAPTER IV

EXPERIMENTAL RESULTS AND ANALYSIS

This chapter contains the results of several runs made with the computational system described in Chapter III. For the first input data set employed, the behavior of total quality cost with various inspection policies is analyzed. Next, the effects of parameter values on optimal solutions is discussed. The third section of this chapter is concerned with the change in optimal inspection policies arising from a different processing cost structure than in the first data set. In the last section the relationships between fixed inspection costs, total quality cost, and optimal policies are examined.

Data Set I - Example

The first set of data to be employed in the model is the following:

$L = 50$ (48 physical stages)

$pn_j = .02,\ j = 2 \ldots 49$

$pr_j = .05,\ j = 2 \ldots 49$

$C_j = .025,\ j = 2 \ldots 49$

$cp_j = .20,\ j = 2 \ldots 49$

$cd_j =-.05j,\ j = 2 \ldots 49$

$cr_j = 2.,\ j = 2 \ldots 49$

$cu2 = 20$ (constant undetected defect cost formulation)

These figures have been chosen to be typical of what might be found in an assembly-line process consisting of many small operations. The electronics industry provides many good examples. To simplify the analysis which follows, stages were assumed to be identical in associated costs and in defect-generating frequencies.

COMPUTATIONAL RESULTS - DATA SET I

Number Of Inspection Points	Stages	TQC
0		19385.939
1 (optimal constrained)	49	4548.006
2 (optimal constrained)	23,49	3774.942
3 (optimal constrained)	15,31,49	3619.872
4 (optimal)	11,22,35,49	3580.167
5	9,18,28,38,49	3580.456
6	7,14,22,31,40,49	3599.244
6a	8,16,24,32,40,49	3599.415
6b	9,17,25,33,41,49	3601.113
6c	9,16,23,34,40,49	3601.996
7	6,12,19,26,33,41,49	3627.288
7a	7,14,21,28,35,42,49	3628.197
8	6,12,18,24,30,36,42,49	3661.669
9	6,11,16,21,26,32,38,42,49	3700.621
9a	5,10,15,20,26,32,38,42,49	3700.691
10	5,9,14,19,24,29,34,39,44,49	3740.068

DATA SET II

2 (optimal constrained)	26,49	12442.688
3 (optimal constrained)	19,34,49	8666.929
18 (optimal)	6,10,13,16,19,22,25,28,31, 33,35,37,39,41,43,45,47,49	4550.824

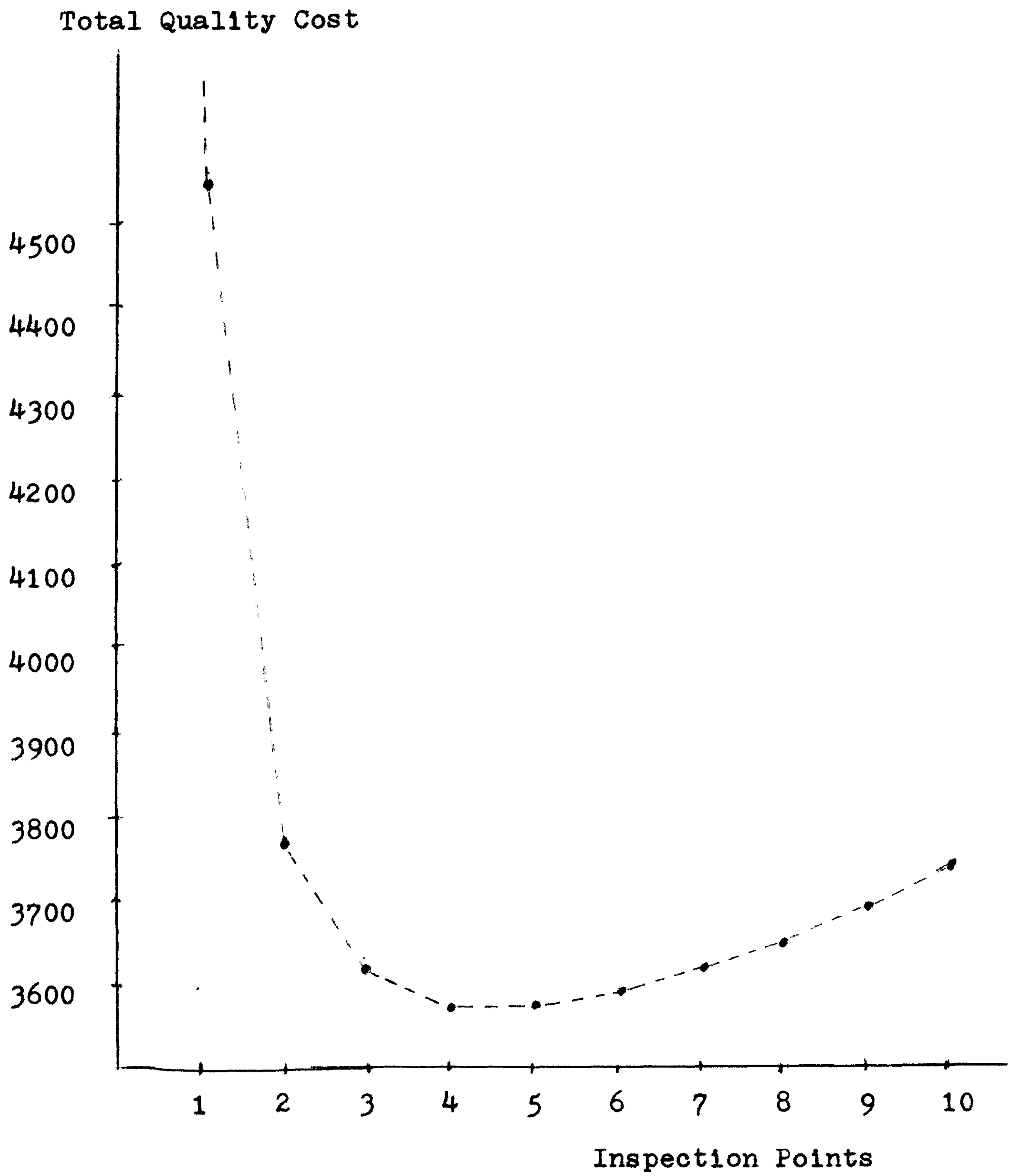


Figure 7 Data Set I Total Cost Curve

The cost of disposal is negative, indicating a positive salvage value of non-repairable units in various states of completion. The salvage value rises linearly with the number of operations having been performed on the unit (if each operation is the addition of a component to an assembly this may be a fairly good approximation). Thus, a non-repairable unit removed from the process at any point will have approximately one-fourth of its cumulative processing cost recovered when scrapped.

The optimal unconstrained, constrained, and arbitrary policies are tabulated on the next page with the total quality cost (TQC) for each case considered. The optimal number of inspection stations was found to be four, distributed very nearly evenly among the 48 stages, with slightly increasing spacing near the end. Total quality cost for this policy is \$3,580.17 for a batch size of 1000. Even as the maximum number of inspection points is constrained to various degrees, the inspection stations continue to be distributed more or less evenly for minimum TQC, as the total cost of course rises. This suggests that an intelligent choice of policies for an arbitrary allocation of more than the unconstrained optimal number (four) of inspection points might be to distribute these stations approximately uniformly. This was done for policies utilizing 5,6,7,8,9, and 10 stations.

Two significant insights can be gleaned from the computational results. The first of these is derived from the behavior of total quality costs as the number of inspection stations is varied. Of course, a very high TQC results from no inspection at all due to the imputed cost of outputting defective units. This cost is equal to cu2 times the expected number of units with a defect of at least one type, $B_1(1 - \prod_{j=2}^{L-1} (1-pn_j+pr_j))$. The cost of \$20 for each defective unit outputted ensures, given the other data, that an optimal policy will require that inspection take place at the last stage.

With the cost data for this example, assuming that there is one or more inspection stations available, the expected cost of undetected defects will rise by $(pn_{L-1}+pr_{L-1})B_{L-2}\cdot ecu2$ if the last inspection point is moved from the last physical inspection stage to the one immediately preceding, L-2. Thus, the cost savings in other cost classes must be of at least this magnitude for terminal inspection at stage L-2 to be economically preferable to terminal inspection at stage L-1. This argument can obviously be extended to terminal inspection at stages L-3, L-4, etc. Although this situation is feasible utilizing appropriate data, it would in most cases necessitate extremely low defect frequencies and low expected cost of undetected defects. We might more generally expect that under a wide range of circumstances found in the real world terminal inspection at stage L-1 will be part of an optimal policy.

Note that for the data employed the total quality cost is a convex function of the number (integer) of inspection stations employed. More significantly, the TQC of n inspection points more than the optimal number is consistently below the cost of n fewer inspection stations, indicating that it will generally be cheaper to err on the high side in assigning inspectors in a shop situation with this data. However, this may not be the case with more perverse parameter values. Here, however, even for a 100% overestimation of the optimal number of inspection points, TQC is only 2% greater than its optimal value.

The second point of interest to be observed from the tabulated results is that, for identical stages, the precise placement of inspection stations appears to be relatively inconsequential. The multiple arbitrary policy solutions for different distributions of the same number of inspection stations yielded approximately the same total quality cost (ex. 6,6a,6b,6c). These alternative distributions were dictated only by the suggestion to "distribute inspection stations approximately uniformly among the stages." In no case is the differential cost between alternative distributions greater than 1/10% of the total cost.

Determinants Of The Solution Obtained

Certain general statements about the optimal solution to the unconstrained inspection problem can be made in many instances. Of particular interest is the behavior of the optimal policy resulting from various changes in the input data set. Although proofs of the following propositions are not provided here, they are in most cases quite obvious considering the structure of the model presented in Chapter II.

1) Total quality cost is a non-decreasing function of each of the cost elements in the model.

2) Ceteris parabis, for some (relatively high) values of the cost of undetected defects (cu1 or cu2) terminal inspection at stage L-1 will be optimal. For all other (relatively low) values terminal inspection at stage n, $n < L-1$, will be optimal. It has previously been suggested that the former will be the case in most actual industrial situations.

3) High probabilities of generating non-repairable defects and high processing costs will both tend to increase the number of inspection points in the optimal unconstrained solution. In particular, high (relative to other stages) values of cp_i for $i > j$ and pn_k for $m < k \leq j$ will tend to require in the least-cost policy inspection at stage j or soon after in order to avoid high processing cost losses.

4) If cr_{jn} for each j is an increasing function of n, the stage at which the type j repairable defect is detected, there will be a tendency for more inspection points in the optimal policy than otherwise, since to defer the detection and repair of defects will increase the repair cost once the defect is detected.

5) As the costs of disposal of defective units, cd, are made more negative, the optimal policy will tend to require fewer inspection stations, for the wasted processing effort expended on non-repairables will be to a greater extent recovered when these items are removed from the process. It is obvious that a sufficient condition for inspection at no more than the final stage is

$$\sum_{j=2}^{L-1} cp_j \leq \sum_{j=2}^{L-1} cdj.$$

However, for positive inspection costs this condition is not at all a necessary one, as will be seen presently.

Note that for the optimal solution to the problem with Data Set I, the four inspection stations were not distributed perfectly uniformly, but that the spacing increased somewhat for stations closer to the end of the line. For identical stages, two primary forces are at work in creating a non-uniform optimal spacing of inspection points. On the one hand, as we proceed down the line, fewer units are to be inspected at each station since non-repairable units have been removed by each preceding inspection. Thus, the variable cost of inspection decreases toward the end of the process, tending to concentrate inspection stations at this end of the line. A decrease in the wasted processing cost also results from the "thinning out" of the batch at stages near the end of the process. This produces a countervailing tendency to have inspection less frequently towards the end of the process. For the particular input data of Data Set I, this effect is apparently the dominant one.

Also of interest are the conditions required for the least-cost policy to require inspection at one stage only. For simplicity, assume for the moment that the cost of undetected defects and defect probabilities are such that terminal inspection at stage L-1, the last stage in the physical process, will be included in the optimal policy. Then a necessary and sufficient condition for inspection at stage L-1 only to be strictly cost-preferable to a policy involving an additional inspection point at any intermediate stage, R, is that the following relation hold for R=2,3...L-2:

$$0 < fic_R + B_1 uic_{1R} + B_R uic_{RL-1} - B_1 uic_{1L-1} + (B_R - B_{L-1})\left(\sum_{j=2}^{R} cr_j pr_j\right)$$
$$+ (B_1 - B_R)\left(cd_R - cd_{L-1} - \sum_{j=R+1}^{L-1} cp_j\right).$$

This result may be derived as follows:

TQC for inspection at stage L-1 only is

(a) $B_1 uic_{1L-1} + fic_{L-1} + erc_{1L-1} + ecs_{1L-1} + ecu(1 \text{ or } 2)_L$

Similarly, TQC for inspection at stage L-1 and some intermediate stage R is

(b) $B_1 uic_{1R} + B_R uic_{RL-1} + fic_R + fic_{L-1} + erc_{1R} + erc_{RL-1} + ecs_{1R} + ecs_{RL-1} + ecu(1 \text{ or } 2)_L$.

If the difference, (b)-(a), is positive for all R, inspection at stage L-1 only is of lesser total quality cost. Subtracting, we obtain the necessary and sufficient condition:

$0 < (b)-(a)$

or

(*) $$0 < fic_R + B_1 uic_{1R} + B_R uic_{RL-1} - B_1 uic_{1L-1} + erc_{1R} + erc_{RL-1} - erc_{1L-1} + ecs_{1R} + ecs_{RL-1} - ecs_{1L-1}.$$

Let us define Δerc to be equal to $erc_{1R} + erc_{RL-1} - erc_{1L-1}$. Then $\Delta erc =$

$$B_1 \left(\sum_{j=2}^{R} cr_j pr_j \right) \prod_{i=2}^{R} (1-pn_i) + B_R \left(\sum_{j=R+1}^{L-1} crj\ prj \right) \prod_{i=R+1}^{L-1} (1-pni) - B_1 \left(\sum_{j=2}^{L-1} cr_j pr_j \right) \prod_{i=2}^{L-1} (1-pn_i).$$

Simplifying, we obtain:

$$\Delta erc = B_R \sum_{j=2}^{R} cr_j pr_j + B_{L-1} \sum_{j=R+1}^{L-1} cr_j pr_j - B_{L-1} \sum_{j=2}^{L-1} cr_j pr_j$$

$$= (B_R - B_{L-1}) \sum_{j=2}^{R} cr_j pr_j .$$

We may also define Δecs to be equal to $ecs_{1R} + ecs_{RL-1} - ecs_{1L-1}$. Then

$\Delta ecs =$

$$\sum_{j=2}^{R} (B_1 - B_{j-1}) cp_j + \sum_{j=R+1}^{L-1} (B_R - B_{j-1}) cp_j - \sum_{j=2}^{L-1} (B_1 - B_{j-1}) cp_j + cd_R (B_1 - B_R) + cd_{L-1}(B_R - B_{L-1}) - cd_{L-1}(B_1 - B_{L-1}).$$

Simplifying, we obtain:

$$\Delta ecs = B_1 \sum_{j=2}^{R} cp_j + B_R \sum_{j=R+1}^{L-1} cpj - B_1 \sum_{j=2}^{L-1} cp_j + cd_R (B_1 - B_R) + cd_{L-1}(B_R - B_1)$$

$$= (B_R - B_1) \sum_{j=R+1}^{L-1} cp_j + (cd_R - cd_{L-1})(B_1 - B_R).$$

$$\Delta ecs = (B_1 - B_R)(cd_R - cd_{L-1} - \sum_{j=R+1}^{L-1} cp_j).$$

Substituting Δerc and Δecs into (*) we obtain the necessary and sufficient condition in a more simplified form:

$$0 < fic_R + B_1 uic_{1R} + B_R uic_{RL-1} - B_1 uic_{1L-1} + (B_R - B_{L-1}) \sum_{j=2}^{R} cr_j pr_j$$
$$+ (B_1 - B_R)(cd_R - cd_{L-1} - \sum_{j=R+1}^{L-1} cp_j).$$

Notice that Δerc becomes simply the expected number of units which become non-repairably defective between stages R and L-1 multiplied by the expected repair cost expended on these units prior to stage R. The expression for Δecs is the expected number of units to become non-repairably defective up to stage R times the net disposal and processing costs incurred from not removing defective units at stage R.

If we assume convexity of the total quality cost as a function of the number of inspection points, optimally assigned, as suggested by computational results (this has not been proven) or even the weaker condition of quasiconvexity of this discrete function, then the relation above suffices for the one inspection point policy to be strictly cost-preferable to any policy involving two or more inspection points and thus globally optimal. This relation does not constitute, however, a short-cut or computationally simple way for an existing production system employing outgoing inspection only to be tested for optimality.

Data Set II - Sensitivity of Solution to Processing Costs

Although the propositions presented in the preceding section can certainly be verified by executing the computational system with appropriate data inputs, such an effort was considered to be of little real practical value in providing additional insights. However, a gross indication of the sensitivity of the optimal solution to changes in processing costs may be of interest. Data Set II is identical to Data Set I but for one exception. Instead of taking the processing cost to be .20 for each stage, processing cost was determined by the relation $cp_j = .05 + .05j$. One might expect from the previous discussions that the optimal policy resulting would have inspection more frequently than before in order to avoid wasting the more expensive processing effort and, additionally, that inspection will tend to be more frequent towards the end of the process than at the beginning, since unit processing costs rise linearly with the stage number.

These expectations were indeed confirmed, as is evident from the information on page . Two interesting results are the much greater number of inspection points required in the optimal solution, and the optimal constrained policies for 2 and 3 inspection stations also requiring closer spacing of inspection towards the end of the line than at the beginning. Unfortunately, a complete cost curve for various degrees of constraint could not be obtained due to the computer's inability to store the number of feasible arc vectors generated by SHORT2 for more than three potential inspection points.

Data Set III - Analysis of Fixed Inspection Cost

The sensitivity of an optimal inspection policy to changes in the fixed cost of inspection stations is investigated through use of the following set of data:

L=12 (10 physical stages)

$pn_j = .02, \quad j = 2..11$

$pr_j = .05, \quad j = 2..11$

$C_j = .025, \quad j = 2..11$

$cp_j = 1.0, \quad j = 2..11$

$cd_j = 0., \quad j = 2..11$

$cr_j = 1.0, \quad j = 2..11$

$cu2 = 20$

Thid data set may represent a flow-shop production process consisting of stages of aggregated elementary operations. The optimal inspection policies and TQC resulting are presented on the next page for equal fixed inspection costs at each stage ranging from \$10 to \$70 per batch.

Over any of the ranges for which the optimal inspection policy remains the same, ΔTQC obviously must equal the number of inspection stations, n, times the change in fixed inspection cost, Δfic. It is observed that as fic increases for all stages, the solution must at some point require a fewer number of inspection points for minimum total quality cost. It appears from the tabulated results that unit decreases in number of inspection points at break-points may not be the rule. Note that as fic is increased to \$20 from \$15 the optimal policy requires a decrease of five inspection points. The exact break-point has experimentally been found to be about \$17.

COMPUTATIONAL RESULTS - DATA SET III

Fixed Inspection Cost	Optimal Policy	TQC
10	2,3,4,5,6,7,8,9,10,11	776.830
15	2,3,4,5,6,7,8,9,10,11	826.830
20	3,5,7,9,11	864.690
30	3,5,7,9,11	914.690
40	3,5,7,9,11	964.690
50	3,5,7,9,11	1014.690
60	3,5,8,11	1055.636
70	3,5,8,11	1095.636

Since $\frac{\Delta TQC}{\Delta fic} = n$, we may postulate that TQC is a concave function of fic since it is piecewise linear with non-increasing slope n. In addition, the results suggest that the optimal policy is relatively insensitive to fic for the data used. Thus, the exact cost figure for inspection is not at all a crucial policy-determining variable. It is increasing to note that for sufficiently large increases in fixed inspection costs, total quality cost will rise by an amount less than the increase in total fixed inspection costs. More significantly, sufficiently large reductions in fic will tend to reduce total quality cost by an even greater amount, indicating a high payoff associated with efforts to improve utilization of inspectors. Thus, a decision to increase the

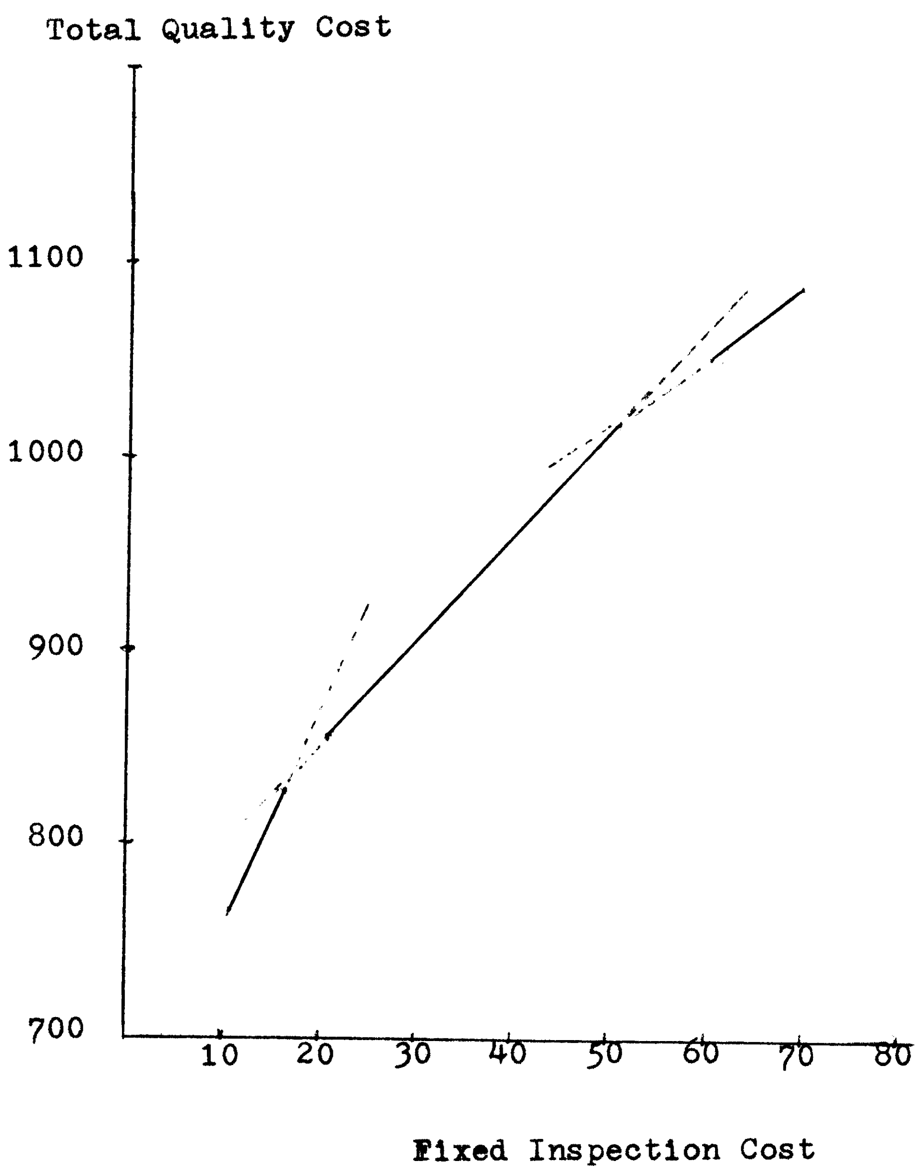


Figure 8 TQC vs. Fixed Inspection Cost

efficiency of inspection through a reduction of either the total time required for maintaining the inspection station for processing one batch or the cost per unit time of inspection should be based on the total savings possible including that resulting from a reoptimized inspection policy. This extra cost reduction may be of considerable magnitude if significant changes in the optimal distribution of inspection points occur at break-points, such as the addition of five inspection points observed as fic is decreased from $20 to $15 in the example above.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE STUDY

Reflections On The Model

The mathematical model and computational system described in this thesis represent a feasible method of solving a rather limited class of inspection policy problems under a set of "reasonable" assumptions about the production and inspection processes. The feasibility of the construction presented herein has been demonstrated for problems of a large enough scale to be of some interest. In addition, the possibilities for handling large-scale sequential process inspection problems has been discussed in Chapter III. The usefulness of time-shared computation as a tool when sensitivity analyses and rapid solutions are desired is quite apparent.

The underlying motivation for this thesis has been White's[4] work, from which the simple network formulations first emerged. The extension to the case of multiple defect classes (repairable and non-repairable) at each processing stage, the possibility of multiple defect kinds of each class, and the presentation of an efficient way to include inspection costs are the primary innovations of the present model. Although somewhat restrictive, White's model is a good deal more general than that of Pruzan and Jackson. The latter model considers single defect classes, a minimal cost structure, and essentially identical inspection set assumptions as in the present model, and the optimization problem is cast into dynamic programming recursions which are equivalent to the shortest route formulation of the model of Chapter II.

It should be noted that the network formulation of the present model can be expressed in a dynamic programming format, just as there exist for all discrete dynamic programming problems with linear objective functions equivalent network formulations. The first optimization network will then be represented with an n-1 component unidimensional decision space at each stage n. The decision consists of choosing the stage m at which inspection is to last take place.

In general, a discrete dynamic programming problem of the generic form

$$f_n(s_n) = \min_{d_n \varepsilon D_n(s_n)} \{c(s_n,d_n) + f_{n-1}(T(s_n,d_n))\}$$

with states s and decisions d $\qquad s_n \varepsilon S_n$, n=1...N

can be transformed into a shortest-route network problem by connecting nodes representing all possible states at each stage with directed arcs with transition costs $c(s_n,s_{n-1})$ equal to $c(s_n,d_n)$ above where $s_{n-1} = T(s_n,d_n)$. For the network model presented herein for the unconstrained inspection problem the state is merely the stage, n. The inspection assumption for any state-stage combination (m,n) is that inspection is to take place for all defect types in the set (m+1, m+2, ...n). Thus, the incremental cost value c(m,n) is uniquely determined through specification of the present state n and the unidimensional decision m.

A generalization of this model has been investigated by Lindsay[10] (1968). In his doctoral thesis, Lindsay removes the assumption that the stage n and decision m uniquely determine the inspection set at stage n. Inspection at stage n in the more general case can be over any subset of defect types in the set (2,...n). The implications of this change for the dynamic programming approach are obvious. The decision space is not multidimensional, taking account of the inspection that has to take place preceding stage n and the state space must also include the possible inspection set combinations for stage n. These spaces may be further reduced by physical considerations, as suggested in Lindsay's work, and by dominance and bounding observations. However, it still appears that this approach has computational limitations for problems of large size. Not only is the dynamic programming algorithm in this case feasible only for relatively small problems, but the calculation of the multitude of transition costs required may, as in the present model, require much more computational effort than the optimization algorithm itself.

Promising Areas for Further Investigation

There are several dimensions in which the present model might be extended. The assumption that the defect type inspection set for any inspection point is to consist of all those defect types between m and n is worthy of reconsideration. As previously discussed, however, Lindsay's dynamic programming formulation of the completely general inspection set problem seems to be computationally discouraging.

There may be a formulation of intermediate restrictiveness possible, however, which would be more promising. It appears that a primary advantage of the more general inspection set formulation is to disallow an inspection policy requiring inspection for a defect type at a point so far down the production line that inspection is physically impossible. Such a solution, if obtained using the model described in this thesis, would then be infeasible in the real world.

The present model could be expanded to avoid infeasibilities of this sort by considering the variable cost of inspection to be a function of the location of the inspection point. Thus, the double subscripted variable C_{jn} would be set equal to +infinity for any stage n at which inspection for type j defects cannot be undertaken. However, this formulation would not in general yield the true optimal inspection policy, since it would also preclude inspection for defect types $i \neq j$ at stage n.

If there are relatively few defects types j which have the property that they cannot be tested or inspected for beyond a certain point in the production process, it is possible that a relatively simple dynamic programming formulation can be devised in which inspection is assumed to occur at stage n for all defect types i, $m < i \leq n$, except for those defect types for which inspection at n is infeasible and inspection at an earlier stage for these defect types only may require to take place. An investigation along these lines might produce a more useful model, but one that is still computationally feasible with limited decision and state spaces. One which admits to solution with more powerful mathematical techniques would be especially useful.

Another possible area of interest not yet given significant attention is the extension of the model to production processes involving partition and assembly operations. It is not in general true, for example, that optimizing inspection policies in the primary- and sub-branches of a process including assembly operations will yield a globally optimal policy. It is instructive to note that the interaction of an operation consisting of one or more subassemblies being added to the main assembly can be represented through use of the concept of multiple defect kinds introduced in Chapter II. The sub-assembly at the end of its production branch and at the point of juncture with the main production line may possess repairable and non-repairable defects of the various types possible from its production line branch. These defect types then become the various defect kinds of the assembly stage on the main production line. A possible approach to this problem might entail the use of a decomposition principle involving the iterative soltuion of main and sub-branch inspection problems independently and the transfer of new defect probability and cost information through the assembly stage interface before each new iteration.

As a final possible area for future study, consider the problem of integrating process monitoring and control with optimum inspection. It has been assumed in each of the models thus far discussed that the defect-generating process at each manufacturing stage is stationary multinomial. Therefore, the objective has been simply to find that inspection policy which will minimize expected cost. Most actual quality assurance problems, however, are concerned with dynamic defect-generating processes at each stage. Indeed, a major industrial problem is the early detection and correction of the manufacturing process which from time to time produces excessive numbers of defective goods. In the multistage static production model inspection at the optimal stages provides potentially useful information about the present state of the performance of each manufacturing stage. This information can be used to correct out-of-tolerance performance at some cost, to provide new optimal inspection policies as parameters change, or both. The objective in this case is to minimize expected cost per unit, while maintaining reasonable stability of the system and meeting demand requirements for non-defective finished goods.

The foregoing suggests that a model for the dynamic process above might be fundamentally of a feedback nature, with changes in system parameters producing actions tending to return the system to some equilibrium state. The Industrial Dynamics technique may be of use here in devising good rules for reoptimization and process correction decisions. Future work on this more realistic problem will probably be most useful of all.

In conclusion, a primary purpose of this thesis has been to obtain some useful insights into the problem of inspection in a multistage production process. The work here has necessarily been limited in both scope and depth in order to achieve a computationally feasible, though admittedly simple, model for obtaining solutions. It is hoped that future investigations will significantly extend the present state of analysis of this problem in some or all of the directions suggested.

REFERENCES

1. Lindsay, G.F., and Bishop, A.B. "Allocation of Screening Inspection Effort - Dynamic-Programming Approach", Management Science. 10:2, 1964.

2. White, L.S. "The Analysis of a Simple Class of Multistage Inspection Plans", Management Science. 12:9, 1966.

3. Pruzan, P.M., and Jackson, J.T. "A Dynamic Programming Application in Production Line Inspection", Technometrics. 9:1, 1967.

4. White, L.S. Shortest Route Models for the Allocation of Inspection Effort on a Production Line. MIT-Sloan School Working Paper, 1967 (To Appear in Management Science).

5. Price, W.H. "Least-Cost Testing Sequence", Journal of Industrial Engineering. 10:4, 1959.

6. Derman, C. "The Sequencing of Tests", Columbia Engineering Quarterly. November, 1962.

7. Johnson, S.M., Optimal Sequential Testing. Rand Research Memorandum RM-1652, 15 March 1956.

8. Ford, L.R. Network Flow Theory. Rand Research Paper P-923, 14 August 1956.

9. Dantzig, G.B. Linear Programming and Extensions. Princeton, 1963.

10. Lindsay, G.F. Minimum-Cost Inspection Programs for Multiple Defects. Doctoral Thesis, Department of Industrial Engineering, The Ohio State University, 1968.

APPENDIX

COMPUTER PROGRAMS

This section contains the statement listings of the FORTRAN IV computer program set used to obtain solutions to the inspection problem considered in this thesis. Input and output is from the appropriate pseudo-tape files in all cases. Brief program descriptions follow.

MASTER. This program calculates the expected number of good or repairable units, B_m, to leave each production stage. An initial batch size of 1000 units provides good scaling for all of the programs to follow. The computation uses the simple recursive relation $B_m = B_{m-1}(1-pn_m)$.

ERC. This program will calculate expected cost of repair, erc_{mn}, for all feasible m and n. For each value of n, m is first set to n-1 and the sum and product terms, each with one element, are multiplied together and with B_m to become erc_{mn}. The value of m is then decremented and new sum and product elements are each time combined with the sum and product terms to produce new sum and product terms. When m becomes its lowest value, 1, n is incremented by one and the process repeats.

ECS. This program will calculate the expected cost of scrappage, ecs_{mn}, for all feasible m and n. The wasted processing cost term is straightforward, calculated exactly as per the formula in Chapter II, and this is added to the cost of disposal for non-repairable units discovered at stage n.

EIC. This program will calculate the expected inspection cost, eic_{mn}, for all feasible m and n. First all stage numbers are arranged into a vector INDX in increasing order of the ratio C_j/pn_j. For each m,n pair this list is scanned for the first stage number, k, which lies between m and n. The cost C_k is then assigned to uic_{mn}. The next such stage number, p, that is found adds $(1-pn_k)C_p$ to the present temporary value of uic_{mn}. This process is continued for each m,n pair according to the formula for uic until the whole list has been scanned. The final value for uic_{mn} is then multiplied by the expected number of units to be inspected, B_m, and fic_n is added to the result to yield eic_{mn}. No attempt is made to save the optimal ordering of inspection for each m,n pair, since only the orderings for the arcs on the optimal path are of interest, and these orderings can easily be calculated after the optimal solution is obtained using the simple ratio inspection sequencing theorem.

ECU1. This program will calculate the expected cost of undetected defects, $ecul_{mL}$, for all m less than L under the additive defect cost assumption. First, $ecul_{1,L}$ is calculated to be $\sum_{j=1}^{L} (pn_j+pr_j)cul_j$. Successive terms $ecul_{mL}$, m=2..L-1, simply require the subtraction of $(pn_m+pr_{m-1})cul_m$ from the sum remaining, so calculation is not redundant.

ECU2. This program will calculate the expected cost of undetected defects, $ecu2_{mL}$, for all m less than L under the constant cost of outputting defects assumption. First the probability of a unit not being defective with no inspection at all is calculated according to the formula $p_1 = \prod_{j=2}^{L-1} (1-pn_j-pr_j)$. The expression $B_1(1-p_1)cu2$ would then be the expected cost of undetected defects for this policy. The probability of a unit last inspected at stage 2 not being defective would then be $p_1/(1-pn_2-pr_2)$. Thus the probabilities p_m for each stage m are obtained by successively dividing p_{m-1} by $(1-pr_m-pn_m)$.

The expected cost of undetected defects for terminal inspection at any stage m is $B_m(1-p_m)cu2$.

<u>AGGREG</u>. This program will add the matrices of values of ecs, erc, eic, and ecu1 or ecu2 to yield the matrix of arc-costs, C, for use by the optimization algorithms. The user is asked through a console printout whether he desires the constant or additive undetected defect cost formulation. The first character of the user's response (c or a) is read and tested in order to determine whether the values of ecu1 or ecu2 are to be included in C. If an incorrect character is typed on the console the user is asked to retype his response. A typical console printout is presented following the program listing.

<u>SHORT</u>. This program contains the Ford Algorithm for the unconstrained inspection problem. The matrix of arc-costs, C, is first read into storage. In Phase I of this program the distance to all nodes but the first is set equal to a very large number (99999999.). In Phase II, the "forward pass", the distance to each node j is set equal to y_j^*, where $y_j^* = \min_{i<j} (c_{ij} + y_i^*)$. The value of y_j^* is thus the shortest distance from node 1 to node j. Phase III, the "backward pass", identifies the nodes which lie on the shortest path from node 1 to node L. The numbers of these nodes are the optimal inspection point stage numbers.

Starting from node L, the next lowest numbered node j on the shortest path is identified by the expression $y_L^* - y_j^* - c_{jL}$ being equal to zero. The process is then repeated at stage j and all other nodes which satisfy the relation above. Due to machine truncation errors, the relation above is required to be satisfied only within a tolerance of $\pm$,001 for the next lowest optimal stage number to be identified. The optimal inspection points, total quality cost, and shortest distance to each node is printed on the console. Sample console output follows the program listing.

SHORT2. This program will calculate the optimal placement of inspection points for the constrained inspection problem. In addition to the matrix of arc costs, the program requires as input from the console the maximum number of inspection points, NSPEC. Phase I of this program generates feasible path identifiers (4-tuples) of the form ((L1,K1),(L2,K2)) where L1 and L2 represent adjacent"levels" of the network form presented in Chapter II and K1 and K2 represent stage numbers for each potential inspection station. These 4-tuples are stored into their order of generation in the form of separate vectors.

Phase II sets the distance from origin to each node (the y_j) to a very large value. Next the distance to each node j is reset to $\min_{i \varepsilon S(j)} (c_{ij} + y_i^*)$ where y_i^* is the minimum distance to node i and S(j) is the set of nodes connected by directed arcs to node j. This can be done by a simple simultaneous scan down the vectors containing L1, K1, L2, and K2 since the feasible arc generator routine (Phase I) produces feasible path identifiers in increasing order first of level, and in increasing order of stage number within levels. The arc cost assigned to any arc is simply $c_{K1,K2}$ regardless of the level identifiers. This algorithm segment is denoted as Phase III.

Phase IV identifies the nodes lying on the optimal (shortest) path through the difference technique used in SHORT. The optimal node level identifiers are suppressed in the console output, so the optimal stage numbers only are printed along with total quality cost, as may be seen in the sample console output following the program listing.

LONG. This program will calculate the total quality cost for any arbitrary inspection policy. The user is asked to type on the console the stages at which inspection points are to be located, and the program will add the costs of the arcs connecting each of these node numbers. Although the entire cost matrix, C, is available to the user to perform this simple operation manually, if desired, the time to output and difficulty of manually identifying the desired arc costs for each arbitrary policy make apparent the value of this program. For the 50-node problem there will exist over 1,000 such arc costs. Sample console output follows the program listing.

print master madtrn
W 2023.3

MASTER

MASTER MADTRN 04/02 2023.3

```
C THIS PROGRAM SEGMENT CALCULATES B(M), M=1..L-1
      DIMENSION PN(55), B(55)
      READ (2,18) L
18    FORMAT (I4)
      REWIND 2
      LS=L-1
      READ (4,17) (PN(I), I=2,LS)
      REWIND 4
      B(1)=1000.
      X=1000.
      DO 16 J=2,LS
C THE RELATIONSHIP IS B(M) = B(M-1)*(1-PN(M))
      X=(1.-PN(J))*X
16    B(J)=X
C OUTPUTTING OF B ONTO TAPE 20
      WRITE (20,17) (B(I),I=1,LS)
      REWIND 20
17    FORMAT (F12.3)
      END
```

R 1.250+.366

ERC

```
print erc madtrn
W 2005.0

  ERC MADTRN      04/02  2005.7

      C THIS PROGRAM WILL CALCULATE THE EXPECTED COST OF REPAIR, ERC(
            DIMENSION B(50),CR(50),PR(50),PN(50),ERC(50,50)
            READ (2,30) L
      30    FORMAT(I4)
            LS=L-1
            LA=L-2
            READ (20,31) (B(I), I=1,LS)
      31    FORMAT (F12.3)
            READ (4,31) (PN(I),I=2,LS)
            READ (5,31) (PR(I),I=2,LS)
            READ (9,31) (CR(I), I=2,LS)
            REWIND 2
            REWIND 4
            REWIND 5
            REWIND 9
            REWIND 20
      C INCREMENT N.
            DO 35 N=2,LS
      C SET UPPER VALUE OF K.
            K=N
            BX=1.
            A=0.
      C CALCULATION OF SUM TERM = SUM+(CR(K)*PR(K)) FOR K=N,N-1,..2
      33    A=A+(CR(K)*PR(K))
      C CALCULATION OF PRODUCT TERM=PROD*(1.-PN(K)) FOR K=N,N-1,..2
            BX=BX*(1.-PN(K))
      C LET M=K-1
            M=K-1
      C CALCULATION OF ERC(M,N)=B(M)*(SUM TERM)*(PRODUCT TERM)
            ERC(M,N)=B(M)*A*BX
      C DECREMENT K
            K=K-1
      C TEST FOR LOWEST VALUE OF M
            IF(M.GT.1)GO TO 33
      35    CONTINUE
            DO 67 M=1,LA
            K=M+1
      C OUTPUTTING OF ERC ONTO TAPE 23
      67    WRITE (23,31) (ERC(M,N),N=K,LS)
            REWIND 23
            END
R 1.800+.933
```

print ecs madtrn
W 2016.8

ECS

ECS MADTRN 04/02 2016.8

```
C THIS PROGRAM WILL CALCULATE THE EXPECTED COST OF SCRAPPAGE AND
C WASTED PROCESSING EFFORT, ECS(M,N).
      DIMENSION B(50), ECS(50,50),CP(50),CD(50)
      READ (2,30) L
      LS=L-1
      LA=LS-1
30    FORMAT (I4)
      READ (20,31) (B(I), I=1,LS)
      READ (7,31) (CP(I), I=2,LS)
      READ (8,31) (CD(I), I=2,LS)
31    FORMAT (F12.3)
      REWIND 2
      REWIND 20
      REWIND 7
      REWIND 8
      DO 25 N=2,LS
      NA=N-1
      DO 25 M=1,NA
      X=0.
      MB=M+1
      DO 23 J=MB, N
C CALCULATION OF WASTED PROCESSING COST
23    X=X+CP(J)*(B(M)-B(J-1))
C CALCULATION OF ECS(M,N)=WASTED PROCESSING COST AND COST
C OF DISPOSAL TIMES THE EXPECTED NUMBER OF DEFECTIVE ITEMS
C OCCURRING BETWEEN STAGE M AND STAGE N.
25    ECS(M,N)=X+CD(N)*(B(M)-B(N))
      DO 67 M=1,LA
      K=M+1
C OUTPUTTING OF ECS ONTO TAPE 25
67    WRITE (25, 31) (ECS(M,N), N=K,LS)
      REWIND 25
      END
```

R 1.483+.550

print eic madtrn
W 2012.4

EIC

EIC MADTRN 04/02 2012.5

```
C THIS PROGRAM CALCULATES THE EXPECTED INSPECTION COST EIC(M,N)
C OF HAVING INSPECTION AT STAGE N GIVEN THAT INSPECTION LAST
C TAKES PLACE AT STAGE M. THESE COSTS ARE CALCULATED USING THE
C LEAST COST INSPECTION SEQUENCE FOR EACH SET (M+1,...,N)OF DEFEC
C INSPECTED FOR AT N.
      DIMENSION EIC(50,50), B(50),INDX(50),FIC(50), C(50),PN(50)
      DIMENSION R(50)
      READ (2,16) L
      REWIND 2
16    FORMAT (I4)
      LS=L-1
      READ (6,15) (C(I), I=2,LS)
      REWIND 6
      READ (4,15) (PN(I), I=2,LS)
      REWIND 4
      READ (20,15) (B(I), I=1,LS)
      REWIND 20
      READ (12,15) (FIC(I), I=2, LS)
      REWIND 12
15    FORMAT (F12.3)
C CALCULATION OF RATIOS C(I)/PN(I)FOR ALL STAGES I, CREATION
C OF VECTOR INDX OF STAGE NUMBERS.
      DO 50 I=2,LS
      INDX(I)=I
50    R(I)=C(I)/PN(I)
      I=0
      LA=L-2
41    I=I+1
         T=999999999.
C ARRANGEMENT OF STAGE NUMBERS, INDX(I), IN INCREASING ORDER OF RA
      DO 48 J=2,LS
      IF(R(J)-T) 47,47, 48
47    T=R(J)
      INDX(I)=J
48    CONTINUE
      KT=INDX(I)
      R(KT)= 999999999.
      IF(I .LE. (L-2)) GO TO 41
C FOR EACH M,N PAIR SCAN VECTOR INDX FOR INDX(I) LESS THAN
C OR EQUAL TO N AND GREATER THAN M.  THE FIRST SUCH STAGE J FOUND
C ASSIGNS INITIAL COST C(J) TO EIC(M,N).  THE NEXT SUCH STAGE K
C (WHICH MUST HAVE THE NEXT LARGEST VALUE OF R FOR STAGES BETWEEN
C N) ADDS COST (1-PN(J))*C(K)  TO EIC(M,N), ETCETERA.
      DO 52 N=2,LS
      NA=N-1
      DO 52 M=1,NA
      KOLD=1
      P=1.
```

```
      PN(1)=0.
      EIC(M,N)=0.
      DO 51 I=1,LA
       IF ((INDX(I) .GT. N).OR. (INDX(I).LE.M))GO TO 51
      K=INDX(I)
      P=P*(1.-PN(KOLD))
      EIC(M,N)=EIC(M,N)+P*C(K)
      KOLD=K
51    CONTINUE
C UNIT INSPECTION COSTS ARE MULTIPLIED BY THE EXPECTED NUMBER OF
C UNITS PASSING THROUGH STAGE N (WHICH IS B(M)), AND THE FIXED
C COST OF INSPECTION AT STAGE N, FIC(N) IS ADDED TO OBTAIN
C EIC(M,N).
      EIC(M,N)=B(M)*EIC(M,N)+FIC(N)
52    CONTINUE
      DO 61 M=1,LA
      K=M+1
61    WRITE (22,15) (EIC(M,N), N=K,LS)
      REWIND 22
62    FORMAT (F12.3)
      END
R 3.150+1.183
```

<u>ECU1</u>

```
print ecu1 madtrn
W 2018.9

  ECU1 MADTRN       04/02  2019.0

      C THIS PROGRAM WILL CALCULATE THE EXPECTED COST OF UNDETECTED
      C DEFECTS ECU1(M,L) FOR ALL M LESS THAN L WHEN INSPECTION
      C LAST TAKES PLACE AT STAGE M FOR ADDITIVE DEFECT COSTS.
              DIMENSION PN(50), PR(50),B(50),CU1(50),ECU1(50,1)
              READ (2,17) L
              LS=L-1
      17      FORMAT (I4)
              READ (10,18) (CU1(I), I=2,LS)
              READ (20,18) (B(M), M=1,LS)
              READ (4,18)(PN(I),I=2,LS)
              READ (5,18) (PR(I),I=2,LS)
      18      FORMAT (F12.3)
              X=0.
      C CALCULATION OF ECU1(1,L)
              DO 43 I=2,LS
      43      X=X+(PN(I)+PR(I))*CU1(I)
              LA=LS-1
      C CALCULATION OF ECU1(M,L)=ECU1(M+1,L)-(PN(M+1)+PR(M+1))* CU1(M+1)
              DO 45 M=1,LA
              ECU1(M,L)=B(M)*X
      45      X=X-(PN(M+1)+PR(M+1))*CU1(M+1)
              ECU1(LS,L)=0.
      C OUTPUTTING OF ECU1 ONTO TAPE 24
              WRITE (24,18) (ECU1(I,L), I=1,LS)
              REWIND 24
              REWIND 20
              REWIND 2
              REWIND 4
              REWIND 5
              REWIND 10
              END
R 1.833+.583
```

print ecu2 madtrn

ECU2

W 2021.0

```
  ECU2 MADTRN       04/02  2021.2

      C THIS PROGRAM WILL CALCULATE THE EXPECTED COST OF UNDETECTED
      C DEFECTS ECU2(M,L) FOR ALL M LESS THAN L WHEN INSPECTION
      C LAST TAKES PLACE AT STAGE M FOR A CONSTANT COST, CU2, FOR EACH
      C DEFECTIVE ITEM SENT OUT REGARDLESS OF THE NUMBER OR TYPES OF DEF
      C IT HAS.
            DIMENSION PN(50), PR(50),B(50),ECU2(50,1)
            READ (2,17) L
      17          FORMAT(I4)
            READ (11,18) CU2
            LS=L-1
            READ (20,18) (B(M), M=1,LS)
            READ (4,18) (PN(I), I=2,LS)
            READ (5,18) (PR(I), I=2,LS)
            REWIND 4
            REWIND 5
             REWIND 11
            REWIND 2
            REWIND 20
      18    FORMAT (F12.3)
            X=1.
            LA=LS-1
      C CALCULATION OF ECU2(M,L)
            DO 42 J=2,LS
      42    X=X*(1.-PR(J)-PN(J))
            DO 43 M=1,LA
            ECU2(M,L)=B(M)*CU2*(1.-X)
      43    X=X/(1.-PR(M+1)-PN(M+1))
            ECU2(LS,L)=0.
      C OUTPUTTING OF ECU2 ONTO TAPE 20
            WRITE (20,18) (ECU2(I,L), I=1,LS)
            REWIND 20
            END
R 1.010+.583
```

print aggreg madtrn
W 2027.4

AGGREG

AGGREG MADTRN   04/02 2027.4

```
C THIS PROGRAM WILL AGGREGATE ECS, ERC, EIC, AND ECU1 OR ECU2
C TO YIELD D(M,N),M=1..N-1, N=2...L.
      READ (2,12) L
12    FORMAT (I4)
      REWIND 2
      LA=L-2
      LS=L-1
      DIMENSION D(50,50),DI(50,50),DII(50,50)
      DO 970 M=1,LA
      K=M+1
C READ-IN OF ECS, ERC,AND EIC
      READ (22,950) (D(M,N),N=K,LS)
      READ (23,950) (DI(M,N), N=K,LS)
970    READ (25,950) (DII(M,N),N=K,LS)
       REWIND 22
       REWIND 23
      REWIND 25
950   FORMAT (F12.3)
      PRINT 25
C QUERY OF USER DESIRED TYPE OF COST OF UNDETECTED DEFECTS
25    FORMAT (53H DO YOU WANT ADDITIVE OR CONSTANT COSTS OF UND
     18H DEFECTS )
60      READ 45 ,J
45    FORMAT (A1)
C IDENTIFICATION OF USER RESPONSE
      IF (J-19071700010 .EQ. 0) MM=24
      IF (J-21210183064.EQ.0) MM=26
C IF UNRECOGNIZABLE RESPONSE RETYPE REQUESTED
      IF (J.NE.21210183064 .AND. J.NE.19071700010) GO TO 70
      GO TO 55
70    PRINT 49
49    FORMAT (17H RETYPE RESPONSE )
      GO TO 60
C READ-IN OF ECU1 OR ECU2
55      READ (MM,950) (D(I,L), I=1,LS)
      REWIND 24
      REWIND 26
      DO 995 M=1,LA
C AGGREGATION OF COST TYPES TO YIELD //D//
      K=M+1
      DO 995 N=K,LS
995   D(M,N)=D(M,N)+DI(M,N)+DII(M,N)
      DO 998 M=1,LS
      K=M+1
C OUTPUTTING OF //D// ONTO TAPE 21
998   WRITE (21,950) (D(M,N),N=K,L)
       REWIND 21
      PRINT 999
999   FORMAT (48H END OF AGGREG. READY FOR TOTL COST CALCULATIO
      END
```

R 2.266+.700

CONSOLE PRINTOUT-AGGREG

```
loadgo aggreg
W 2039.3
EXECUTION.
 DO YOU WANT ADDITIVE OR CONSTANT COSTS OF UNDETECTED
 DEFECTS
c
 END OF AGGREG. READY FOR TOTL COST CALCULATION
  EXIT CALLED. PM MAY BE TAKEN.
R 9.100+1.333
```

SHORT

```
print short madtrn
W 2037.8

 SHORT MADTRN       04/02  2037.9

              DIMENSION Y(50),D(50,50),INSP(50)
                  READ(2,65)L
       65     FORMAT (I4)
              LS =L-1
              DO 4 I=1,LS
                  K=I+1
              READ (21,3)(D(I,J),J=K,L)
       3      FORMAT (F12.3)
       4      CONTINUE
       C THIS IS SHORT
       C FORD ALGORITHM FOR MODEL I -- FEASIBLE ARCS INCLUDE ALL
       C      ARC (I,J), I LESS THAN J
       C INPUT - L=NUMBER OF NODES
       C            MATRIX D, D(I,J) =ARC LENGTH FROM NODE I TO NODE J
              LS=L-1
       C PHASE I - INITIALIZATION
              Y(I)=0.
              DO 5 I=2,L
       5      Y(I)=999999999.
       C PHASE II - FORWARD PASS, DETERMINATION OF NODE VALUES
              DO 60 I=1,LS
              M=I+1
              DO 60 J=M,L
              IF(Y(I)-Y(J)+D(I,J)) 6,6,60
       6      Y(J)=Y(I)+D(I,J)
       60         CONTINUE
              PRINT 68
       68     FORMAT (28H SHORTEST DISTANCE TO NODES /16H NODE  DISTANC
              PRINT 99, (I, Y(I), I=1,L)
       99     FORMAT (I4,2X,F12.3)
              DO 7 I=1,LS
       7      INSP(I)=0
              INSP(L)=1
              J=L
       12     M=J-1
       C PHASE III - BACKWARD PASS, OPTIMAL NODE IDENTIFICATION AND OU
              DO 8 I=1,M
              IF(ABS(Y(I)-Y(J)+D(I,J)).LE. .001) GO TO 79
       8      CONTINUE
       79     INSP(I)=1
              J=I
              IF (J-1) 9,9,12
       9          PRINT 27, Y(L)
       27     FORMAT(//,      15H TOTAL COST IS ,F12.3///)
```

```
      PRINT 30
30    FORMAT(     32H PUT INSPECTORS AT STAGES                    )
      DO 16 I=2,LS
      IF(INSP(I)) 16,16,42
42    PRINT 17,I
17    FORMAT (     I4)
16    CONTINUE
      REWIND 2
      REWIND 21
      END
C END OF SHORT
R 2.01[illegible]+.7[illegible]
```

SAMPLE PRINTOUT-SHORT

```
loadgo short
W 2035.5
EXECUTION.
 SHORTEST DISTANCE TO NODES
 NODE  DISTANCE
    1          0.
    2        144.000
    3        235.540
    4        344.289
    5        464.525
    6        568.967
    7        672.447
    8        787.520
    9        886.903
   10        985.423
   11       1095.636
   12       1095.636

 TOTAL COST IS      1095.636


 PUT INSPECTORS AT STAGES
    3
    5
    8
   11
   EXIT CALLED. PM MAY BE TAKEN.
R 5.733+.766
```

```
print short2 madtrn          SHORT2
W 2043.6

SHORT2 MADTRN       04/02  2043.6

           PRINT 789
      789  FORMAT (52H  FORD ALGORITHM FOR CONSTRAINED INSPECTION PR
          1   58H  ENTER THE MAXIMUM NUMBER OF INSPECTORS, RETURN CA
           READ 4, NSPEC
           DIMENSION D(50,50),Y(50,50), N(2,2500), L1(2500), L2(2500
           DIMENSION K1(2500),K2(2500)
           READ (2,4) L
      4    FORMAT(I4)
           LS=L-1
           DO 5 I =1,LS
           K=I+1
           READ(21,3) (D(I,J),J=K,L)
      3    FORMAT (F12.3)
      5    CONTINUE
           REWIND 2
           REWIND 21
      C THIS IS SHORT2
           REWIND 3
      C FORD ALGORITHM FOR MODEL II - EXTENDED SHORTEST ROUTE
      C INPUT - L=NUMBER OF STAGES
      C          MATRIX D, D(I,J)= ARC LENGTH FROM NODE I TO J
      C          NSPEC = NUMBER OF LEVELS IN NETWORK EXCLUSIVE OF INIT
      C                   TERMINAL NODES
           LP=L-2
           IF (NSPEC.GT.L-2) NSPEC=L-2
           IF (NSPEC .LE. 0) TAP=D(1,L)
           IF (NSPEC .LE. 0) GO TO 500
      C PHASE I - GENERATION OF FEASIBLE PATH VECTORS L1,K1,L2,K2
           DO 90 J=1,LP
           N(1,J)=1001
      90   N(2,J)=2001+J
           JX=2
           IXO=NSPEC
      11   IF(IXO-1) 18,18,14
      14   K=1
      30   DO 15 I=1,K
           N(2,J)=(JX+1)*1000 + JX+K
           N(1,J)=JX*1000+JX+I-1
      15   J=J+1
           IF(L-K-JX-1) 16,16,17
      17   K=K+1
           GO TO 30
      16   JX=JX+1
           IXO=IXO-1
           GO TO 11
```

```
18    INDX =(NSPEC+2)*1000+L
      N(2,J)=INDX
      N(1,J)=1001
      J=J+1
      NS=2
71    NT=NS*1000
      ID=NS
      DO82 I=NS,LS
      N(1,J)=ID+NT
      N(2,J)=INDX
      J=J+1
82    ID=ID+1
      NS=NS+1
      IF(NS-NSPEC-2) 71,83,83
83    NUM=J-1
C FEASIBLE PATH (I) IS (L1(I),K1(I))--TO--(L2(I),K2(I)) WHERE L
C NODE LEVEL AND K IS THE NODE STAGE
      DO 12 I=1,NUM
      I1=N(1,I)
      I2=N(2,I)
      L1(I)=N(1,I)/1000
      L2(I)=N(2,I)/1000
      K1(I)=I1- L1(I)*1000
12    K2(I)=I2-L2(I)*1000
C PHASE II - INITIALIZATION
      DO 55 I=1,NUM
      LA=L2(I)
      LB=K2(I)
55    Y(LA,LB)= 999999999.
      Y(1,1)=0.
C PHASE III - FORWARD PASS, DETERMINATION OF NODE VALUES
      DO 78 I=1,NUM
      IA=L1(I)
      IB =K1(I)
      IC=L2(I)
      ID=K2(I)
      IF (Y(IA,IB)+D(IB,ID)-Y(IC,ID)) 76,78,78
76    Y(IC,ID)=Y(IA,IB)+D(IB,ID)
```

SAMPLE PRINTOUT-SHORT2

```
loadgo short2
W 2036.3
EXECUTION.
  FORD ALGORITHM FOR CONSTRAINED INSPECTION PROBLEM
  ENTER THE MAXIMUM NUMBER OF INSPECTORS, RETURN CARRIAGE
3


 TOTAL COST IS       1096.713



 PUT INSPECTORS AT STAGES
  11
   7
   4
  EXIT CALLED. PM MAY BE TAKEN.
R 6.850+1.666
```

loadgo long

<u>SAMPLE PRINTOUT-LONG</u>

```
W 2037.3
EXECUTION.
 THIS PROGRAM WILL CALCULATE TOTAL COST FOR ANY
 ARBITRARY SET OF STAGES WHERE INSPECTION IS TO TAKE PLACE.
 ENTER STAGE NUMBERS IN ASCENDING ORDER, RETURNING THE
 CARRIAGE AFTER EACH ENTRY, AND END THE LIST WITH
 A ZERO.
3
7
12
0
 TOTAL COST IS       5153.348
  EXIT CALLED. PM MAY BE TAKEN.
R 6.866+1.500
```

print long madtrn
[illegible].2

LONG

LONG MADTRN 04/02 2041.3

```
C THIS PROGRAM WILL CALCULATE TOTAL COST FOR ANY ARBITRARY INSP
C POLICY, WITH STAGE NUMBERS BEING SUPPLIED FROM THE CONSOLE
      DIMENSION D(50,50), N(50)
      READ (2,5) L
      REWIND 2
5     FORMAT (I4)
      LS=L-1
      DO 25 I=1,LS
      K=I+1
      READ (21, 30) (D(I,J), J=K,L)
25    CONTINUE
      REWIND 21
30    FORMAT(F12.3)
      N(1)=1
      I=2
       PRINT 11
11    FORMAT (48H THIS PROGRAM WILL CALCULATE TOTAL COST FOR AN
     100H ARBITRARY SET OF STAGES WHERE INSPECTION IS TO TAKE P
     254H ENTER STAGE NUMBERS IN ASCENDING ORDER, RETURNING THE
     350H CARRIAGE AFTER EACH ENTRY, AND END THE LIST WITH ,/
     4 9H A ZERO. )
C READ-IN OF STAGE NUMBERS FROM CONSOLE
2     READ 109 , N(I)
109   FORMAT (I2)
      IF (N(I)) 3,3,100
100      I = I+1
      GO TO 2
3     KOLD =1
      TC = 0.
      N(I)=L
C CALCULATION OF TOTAL COST=D(1,I)+D(I,J)+D(J,K)...+D(Q,L)
      DO 4 J=2,I
      M=N(J)
      TC = D(KOLD, M)+ TC
4     KOLD = N(J)
      PRINT 120, TC
120   FORMAT (15H TOTAL COST IS , F12.3)
      END
```

R 1.710+.000