1.0

2.8   2.5

3.2   2.2

2.0

1.1   1.8

1.25 1.4 1.6

MICROCOPY RESOLUTION TEST CHART

AD A113424

INFORMATION STRUCTURES

FOR SINGLE ECHELON ORGANIZATIONS[+]

by

Debra A. Stabile[*]

Alexander H. Levis[**]

Susan A. Hall[**]

DTIC
ELECTE
APR 1 4 1982
D

ABSTRACT

A methodology for designing the information structures for decision makers who comprise the boundary between an organization and its environment is presented. The environment is modeled as a source that generates symbols or messages that the organization members must process without being overloaded. Two basic information reduction strategies are considered: 1) creation of self contained tasks, and 2) creation of slack resources. The former leads to the partitioning of the input signal and the parallel processing of the partition; the latter to alternate processing where each decision maker receives signals according to some deterministic rule but is given more time to process them, i.e., a delay is introduced. These two strategies are then integrated to produce a variety of information structures for special cases.

DTIC FILE COPY

INTRODUCTION

An organization, perceived as an open system [1], interacts with its environment: it receives signals or messages in various forms that contain information relevant to the organization's tasks. These signals must be identified, analyzed and then transmitted to their appropriate destinations within the organization. The way in which an organization accepts and processes these signals affects its internal structure and has direct consequences on its performance. The sources of the signals and their properties, the tasks to be performed, and the capabilities and limitations of the individuals comprising the organization are key factors in determining the structure of an organization.

A major simplification occurs when only the boundary between the organization and its environment is considered. While the organization members on the boundary may occupy different positions in the internal organizational structure, their common characteristic is that they receive direct inputs from the environment. In that sense, they constitute a single echelon. However, individuals, or groups of individuals, can have very different capabilities and limitations that reflect, indirectly, their position in the crganization. For example, they can process only certain classes of signals (specialization) or they can deal with limited levels of uncertainty. Since it is important to remember that the single echelon may incluce commanders as well as operators of monitoring systems, executives as well as clerks, the term decision-maker is used to describe all members.

In this paper, a methodology for designing information structures for single echelon organizations is developed. The choice of decision makers (DMs) that comprise the single echelon (SE) and the rules for assigning signals to them define the organizational form.

For the types of organizations considered, the performance of a task is equivalent to the processing of information, where information is defined to be the data received by the DMs in the SE. Galbraith has argued that variations in the amounts of information (data) that are processed are primarily responsible for the variations in organizational forms. [2]

A tacit assumption in this work is that a single DM cannot process the available data while simultaneously achieving the desired performance level. When a DM has been assigned more data than he is able to process in the prescribed time interval, while still maintaining a given performance level, he can react in one of several ways. He may decide to reduce the amount of data he has to process by either randomly (rejection) or selectively (filtering) omitting data. The amount of data he may be required to process may be reduced also by having it preprocessed. He may decide to reduce the number of categories of discrimination, i.e., approximate the inputs, or he may reduce the required level of accuracy for processing the data and, in so doing, reduce the number of different outputs. If these alternatives seem unsatisfactory, he may decide to receive all the data, allowing queues to build up, delaying the processing during periods of peak loads and attempting to catch up during slow periods. Otherwise the DM may simply choose not to perform the task. J. Miller found that at moderate rates of information input overload, all these methods described were used about equally. When the input rate far exceeded a DM's processing capacity, however, random and selective omission were the most significant methods of dealing with the situation. [3].

An alternative to having the data preprocessed is the employment of multiple parallel channels [4]. The concept of parallel DMs is analogous to the idea of distributed information processing with each DM performing a subtask. Many studies in the literature have revealed that as the uncertainty of the tasks increases, the "flatter," i.e., more distributed, an organization should become with respect to its DMs. [4]

Galbraith has suggested two information reduction str   s for organizations to address this issue: (1) Creation of Self-Contained Tasks, and (2) Creation of Slack Resources. [2] In the first strategy, the original task is divided into a set of subtasks. This reduces the diversity of inputs each DM receives as well as the diversity of outputs he must produce. In the second strategy, the slack resource of interest is time.

Thus, the trade-off between performance and delay in accomplishing the task becomes an important design consideration.

Two types of processing modes will be considered, parallel processing, which is associated with the first strategy, and alternate processing, which is associated with the second strategy. These are fundamental strategies as discussed first by Drenick [5], that can be used to process the incoming signals without overloading any DM in the single echelon. The two fundamental modes can be integrated in various ways so as to develop more complex organizational forms.

Parallel processing is introduced in order to reduce the amount of information any particular DM receives. This assumes the task can be divided into subtasks with each subtask requiring some subset (not necessarily mutually exclusive) of the information. The subtasks are selected and assigned to DMs in such a way that each DM is capable of processing his data before his next input is received. This is referred to as parallel processing because the subtasks are carried out in parallel within the same time interval. This processing mode guarantees that the expected delay of any processed input is equal to the mean input interarrival time. The structure of the single echelon with parallel processing is shown in Figure 1; S represents the source, z_i the signal received by the i-th DM and y_i is the output he produces.
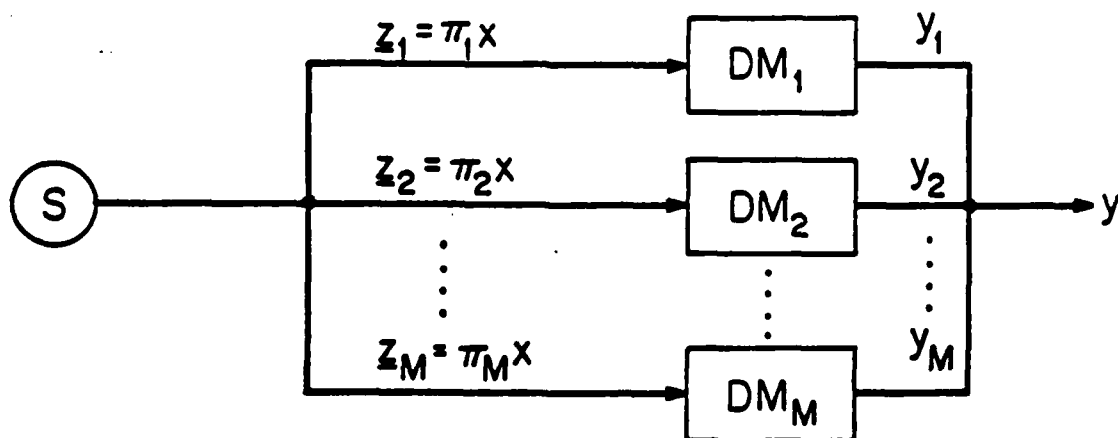

Figure 1. Parallel Processing

-3-

If signals arrive at a rate δ^{-1}, and a DM requires more than time δ to process the information (queueing of information has been removed from further consideration), additional DMs are introduced into the single echelon; each DM is assigned a different input signal. The number of additional DMs must be sufficient to receive and process information so that no DM receives another input signal until the previous one he received has been processed. This is referred to as alternate processing because the assignment of the inputs alternates among the DMs in the SE. The structure of a single echelon with one form of alternate processing is shown in Figure 2. The precise rule for allocating inputs to the various DMs determines the minimum number of DMs necessary to process the inputs without any DM being overloaded.
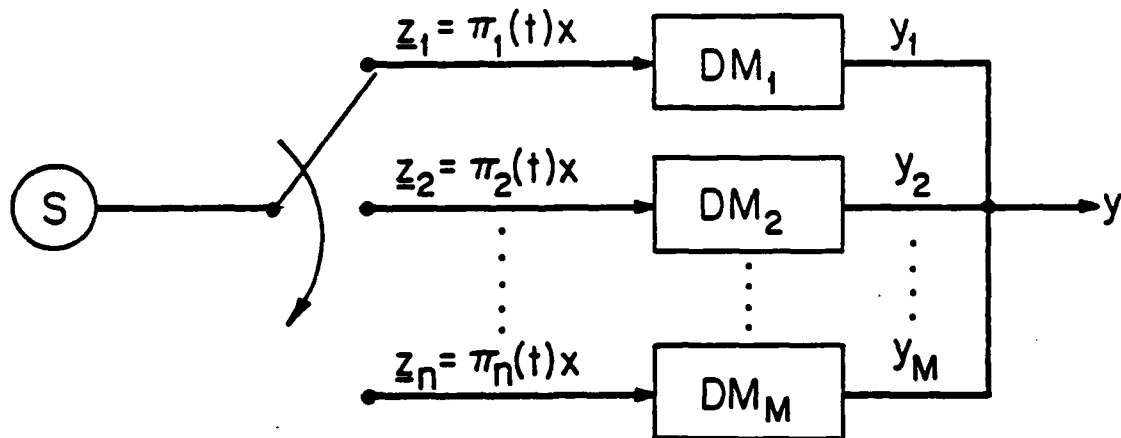
Figure 2. Alternate Processing (Periodic)

In general, parallel processing is associated with the partitioning of the task into subtasks, in the absence of any slack in time. Alternate processing is associated with the allocation of the slack resource (time). The task is not partitioned, but each DM is allotted more time to carry out complete tasks. In both parallel and alternate processing the input rate is equal to the output rate; in alternate processing, however, a delay that is strictly greater than the mean input interarrival time is introduced.

-5-

TASK AND DECISION MAKER MODELS

In this section, two elements of the design problem are modeled: the task to be performed and the properties of the decision makers who comprise the single echelon.

The single echelon is assumed to receive signals from one or more sources external to it. Every δ_n units of time on the average, each source n generates symbols, signals, or messages x_{ni} from its associated alphabet X_n, with probability p_{ni}, i.e.,

$$p_{ni} = p(x_n = x_{ni}) \quad ; \quad x_{ni} \in X_n \quad ; \quad i = 1,2,\ldots,\gamma_n \qquad (1)$$

$$\sum_{i=1}^{\gamma_n} p_{ni} = 1 \quad ; \quad n = 1,2,\ldots,N'$$

where γ_n is the dimension of X_n. Therefore, δ_n^{-1} is the mean frequency of symbol generation from source n.

The task to be performed is defined as the processing of the input symbols x_n by the single echelon to produce output symbols. It is assumed that a specific complex task that must be performed can be modeled by N' such sources of data. Rather than considering these sources separately, one supersource composed of these N' sources is created. The input symbol $\underline{x}'$, may be represented by an N'-dimensional vector with each of the sources represented by a component of this vector, i.e.,

$$\underline{x}' \cong (x_1,\ldots,x_n,\ldots,x_{N'}) \quad ; \quad \underline{x}' \in X \qquad (2)$$

To determine the probability p_j that vector $\underline{x}'_j$ is generated, the independence between components must be considered. If all components are mutually independent (see, e.g. [5]) then p_j is the product of the probabilities that each component of $\underline{x}'_j$ takes on its respective value from its associated alphabet:

$$P_j = \prod_{n=1}^{N'} P_{nj}.$$
(3)

When all components of the input vector are mutually independent this is referred to as being of <u>finest grain</u>. In many situations, this assumption is unrealistic; it is more common to have some components probabilistically dependent.

If two or more components are probabilistically dependent on each other, but as a group are mutually independent from all other components of the input vector, then these dependent components can be treated as one new supercomponent with a new alphabet. Then a new input vector, $\underline{x}$, is defined, composed of the mutually independent components and these supercomponents. This new $\underline{x}$ is of finest grain.

This model of the sources implies synchronization between the individual source elements so that they may be treated as one input vector. Specifically, it is assumed that the mean interarrival time for each component δ_n is equal to δ.

Finally, it is assumed that every component of each input vector must be processed by at least one DM. If the objective is to minimize the number of DMs necessary to process the input vector, then there is no advantage to having a component processed more than once, since this could require additional DMs. Alternatively, reliability considerations may require that some redundancy in processing be present.

In general, each of the decision makers in the single echelon processes some subset of the components comprising the input vector. Each DM is distinguished by

- a processing time function, $\bar{\tau}$, which yields the mean time for processing a particular set of components,

- his specialization; i.e., which components he is able or qualified to process, and

- a cost function.

A DM is said to be overloaded when the time required for him to process the

components he is assigned exceeds δ, the mean interarrival time of input symbols requiring processing.

The uncertainty of the input symbol generated and the number of possible input-output pairs are two of the factors that affect the mean processing time. Miller introduced a processing time function which has an information theoretic interpretation while Hyman and Hick provided experimental evidence that information, rather than the number of inputs or outputs, was a more appropriate measure [3]. In order to express analytically the mean processing time, it is necessary to describe first the partitioning of the input vector. Let the vector $\underline{x}$ be partitioned into groups of components and let the k-th partition be denoted by an s_k-dimensional vector $\underline{z}_k$. The k-th partition is derived from the input vector $\underline{x}$ using the partitioning matrix π_k, i.e.,

$$\underline{z}_k = \pi_k \underline{x} \tag{4}$$

where π_k is of dimension $s_k \times N$ and rank s_k. Each column of π has at most one non-zero element (unity) while each row has exactly one non-zero element (unity). Since the order of the components in $\underline{z}_k$ is of no consequence, any other matrix obtained by interchanging the rows of π_k yields the same partition.

The partitioning matrix assigned to the m-th DM, π_k^m, specifies the set of symbols that he must process. The information structure, Π, for a single echelon consisting of M DMs is defined by the set of M partitioning matrices associated with the DMs.

The mean processing time for the m-th DM who has been assigned partition k is defined to be:

$$\bar{\tau}_k^m = t^m + c^m H_k. \tag{5}$$

where H_k is the entropy associated with the set of components $\underline{z}_{kj}$ specified by the partition matrix π_k, i.e.,

$$H_k = - \sum_j p(\underline{z}_{kj}) \log p(\underline{z}_{kj}) \qquad (6)$$

and where t^m and c^m are parameters characterizing the particular DM. Since it has been assumed that all components in the input vector are mutually independent, the entropy of $\underline{z}_k$ is equal to the sum of the entropies of each of the components of $\underline{z}_k$.

The expression of the mean processing time (5) may be obtained by averaging, over all elements j in the k-th partition, the symbol processing time given by

$$\tau_{kj}^m = t^m - c^m \log p(\underline{z}_{kj}) \qquad (7)$$

While the model for the mean processing time (5) is a plausible one and is consistent with experimental data, the inferred model for the individual symbol processing times is not (for a discussion, see [3]).

Consider now the M decision makers that are available to the organization designer. These DMs can be grouped in several ways according to their mean processing time function (5), their specialization and their cost.

In the simplest case, all DMs are identical. This implies the same mean processing time,

$$\bar{\tau}_k^m = \bar{\tau}_k = t + c \, H_k \qquad \text{for all m}$$

no specialization, and equal cost.

Groups of identical DMs are considered next. Decision makers within a specified group, $\mathbf{M}^g$, $g = 1,2,\ldots,G$, possess identical processing time functions, are able to process the same types of components and have identical costs.

Decision makers are often experts or specialists in a particular area of an overall task with an attendant reduction in the average processing times for tasks in the given area. Another form of specialization is the

use of machines which are often limited with respect to the types of data they can process. Specialization can be thought of in the context of this work as a constraint that allows the DM to access and process only certain components of data.

Let G groups of DMs exist with identical DMs within each group M^g, and let each of these groups be specialized so that it can be assigned only components from a set of components L^g, $g = 1,...,G$. The cost of each DM depends on the group he belongs to and his area of specialization.

The input vector has been constructed so that its components are independent. These components can be grouped together in terms of several attributes. Components can be grouped so that within each group components have alphabets of equal size and identical probability distributions. Groups can be formed of components that must be processed together, even though they are independent. Another attribute is dependent on the DMs and their specialization. If each DM can process only certain alphabets or even subalphabets, then the components (and their alphabets) can be grouped together according to which DM can process them.

At this point, it is possible to state the first two steps in the design of information structures.

STEP 1: <u>Task</u>

    a) Construct a single supersource;

    b) Restructure input vector so that its components are mutually independent;

    c) Identify groups of components with alphabets that have identical probability distributions;

    d) Identify components that must be processed together;

    e) Identify alphabets and subalphabets that can be processed by specialized DMs.

STEP 2: <u>Decision Makers</u>

    a) Group DMs according to their processing time functions;

    b) Group DMs according to their specialization.

Once these steps have been carried out, the problem of designing the information structure for a single echelon can be formulated. Parallel processing will be considered first.

$$H_k = - \sum_j p(\underline{z}_{kj}) \log p(\underline{z}_{kj}) \tag{6}$$

and where t^m and c^m are parameters characterizing the particular DM. Since it has been assumed that all components in the input vector are mutually independent, the entropy of $\underline{z}_k$ is equal to the sum of the entropies of each of the components of $\underline{z}_k$.

The expression of the mean processing time (5) may be obtained by averaging, over all elements j in the k-th partition, the symbol processing time given by

$$\tau_{kj}^m = t^m - c^m \log p(\underline{z}_{kj}) \tag{7}$$

While the model for the mean processing time (5) is a plausible one and is consistent with experimental data, the inferred model for the individual symbol processing times is not (for a discussion, see [3]).

Consider now the M decision makers that are available to the organization designer. These DMs can be grouped in several ways according to their mean processing time function (5), their specialization and their cost.

In the simplest case, all DMs are identical. This implies the same mean processing time,

$$\bar{\tau}_k^m = \bar{\tau}_k = t + c \, H_k \qquad \text{for all m}$$

no specialization, and equal cost.

Groups of identical DMs are considered next. Decision makers within a specified group, $\mathbf{M}^g$, $g = 1,2,\ldots,G$, possess identical processing time functions, are able to process the same types of components and have identical costs.

Decision makers are often experts or specialists in a particular area of an overall task with an attendant reduction in the average processing times for tasks in the given area. Another form of specialization is the

PARALLEL PROCESSING

In a parallel processing structure, partitions of the input symbol
are selected and assigned to the decision makers (DMs) in the
organization. The group of DMs who, together, process the entire input
symbol form the single echelon (SE). Each DM is constrained to process
a partition of components from those that do not overload him, i.e., from
those that result in a mean processing time of δ or less.

Mathematical programming is an appropriate modeling approach for
this class of problems. This approach seeks "the optimum allocation
of limited resources among competing activities under a set of constraints
imposed by the nature of the problem being studied." [6] In this
context, the components of the input vector correspond to the limited
resources, the DMs correspond to the competing activities and the
constraint sets include processing time capabilities and specialization
limitations of each DM.

Explicit enumeration of all distinct partitions of components
results in a problem with very high dimensionality. Fortunately,
the mutual independence of the components of the input vector allows
an alternative formulation which reduces significantly the size of
the problem. This is a consequence of not having the distinct partitions
enumerated explicitly. Rather, only the components and the DMs need
be considered explicitly.

In the implicit formulation, individual components are selected
and assigned to each DM in the model. The group of components
assigned to the m-th DM defines the partition vector. In this way,
the required M partition vectors are constructed implicitly. The
conditions for selecting and assigning components are

     a)   every component is processed, and

     b)   no DM is overloaded.

Let Y_{nm} be a binary variable which equals one if the n-th component is assigned to the m-th DM, zero otherwise. To guarantee that every component is processed once and only once, the following set of constraints is established:

$$-\sum_{m=1}^{M} Y_{nm} = -1 \qquad\qquad n = 1,2,\ldots N \qquad\qquad (8)$$

where

$$Y_{nm} = 0,1 \qquad\qquad\qquad\qquad (9)$$

A network structure which links every component to every DM is shown in Figure 3.

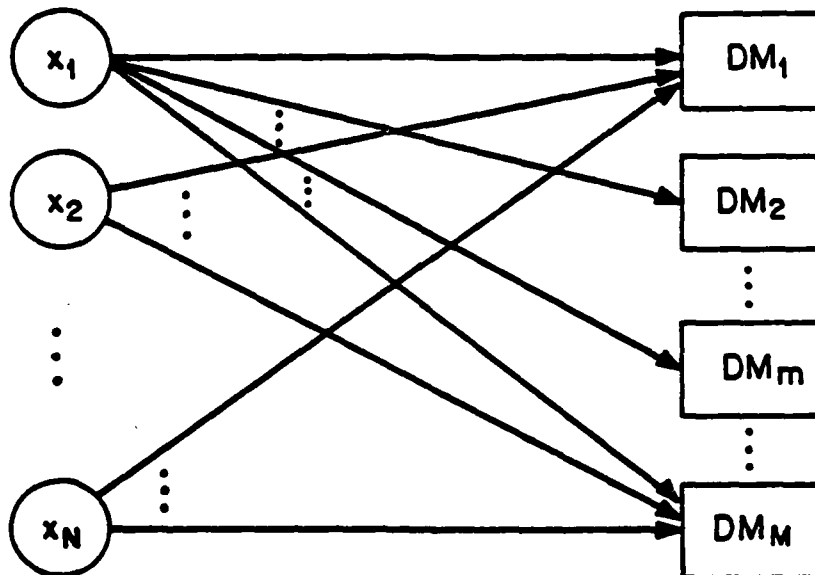**COMPONENTS**   **DECISION MAKERS**

Figure 3.   GN Formulation: Implicit Enumeration

A DM will not be overloaded, if the average time he requires to process the components assigned to him does not exceed δ. The mean processing time (5) is assumed to be given by

$$\bar{\tau}^m = st^m + c^m H \tag{5'}$$

where s is the number of components assigned to the DM and (t^m, c^m) are parameters that characterize the m-th DM. Since the components have been assumed to be mutually independent, the entropy H is equal to the sum of the entropies of the s components. Since the components assigned to the m-th DM are not known a priori, a binary indicator variable Y_{nm} is introduced which includes the time for processing component x_n only if it is assigned to the DM:

$$(t^m + c^m H_n) \; Y_{nm} = \begin{cases} (t^m + c^m H_n) & \text{if } Y_{nm} = 1 \\ \\ 0 & \text{if } Y_{nm} = 0 \end{cases}$$

Furthermore:

$$\bar{\tau}^m = \sum_{n=1}^{N} (t^m + c^m H_n) \; Y_{nm} \leq \delta \qquad m = 1, 2, \ldots, M \tag{10}$$

i.e., $\bar{\tau}^m$ must be less than or equal to δ to guarantee the m-th DM is not overloaded.

The coefficients of the variables Y_{nm} in constraint set (10) are clearly not restricted to unity or zero, implying that the formulation is not a pure network. It is a generalized network (GN), however, since each variable appears in at most two constraint equations. The sets of constraints comprising this GN formulation are given by (8),(9) and (10).

The objective function for this problem that is to be minimized is the number of decision makers required to process all the components without overload.

The information structure can be constructed from the optimal solution to this problem.

Partitioning matrices, π_k^m, corresponding to the set of components x_n for which $Y_{nm} = 1$, for each m, are inferred and then combined to define the information structure:

$$\Pi = (\pi_{k_1}^1, \ldots, \pi_{k_m}^m, \ldots, \pi_{k_M}^M)$$

In the following sections, several special cases are presented to illustrate the solution procedures for this class of problems.

Single Group of Identical DMs

All components are assumed one-dimensional and mutually independent and all DMs possess identical properties. These simplifying assumptions do not lead to a reduction in the dimensionality of the GN, however. Each DM must be considered separately and his particular assignment of components obtained. This knowledge would be lost, if a single representative DM was used instead.

There is an alternative approach, however, to solving the problem of selecting and assigning components to all DMs simultaneously, which takes advantage of these simplifying assumptions. This approach consists of sequentially solving M^* MP problems belonging to a class known as knapsack problems. A knapsack problem (KP) is composed of exactly one constraint and variables which are all binary. The KP formulation for the m-th problem is:

$$\tau_n^{-m} Y_{mn} \leq \delta \qquad m = 1, 2, \ldots, M^*; \quad n \in L^m$$

where L^m is the set of components available for assignment; i.e., not assigned in one of the previous (m-1) KPs. The set L^m has dimension s_m where

$$s_m = N - \sum_{m'=1}^{m-1} s_{m'} \qquad m = 1, 2, \ldots, M^*$$

The first KP assigns as many of the N components as possible (s_1) to a DM. The second KP assigns as many of the remaining ($N-s_1$) components as possible to another DM.

This process is repeated until all N components are assigned. Very efficient algorithms using Branch and Bound techniques exist to solve these KPs. [3] This leads to the possibility that the M^* problems, each with ($1+s_m$) variables and one constraint, may require cumulatively less total computer time than solving one GN problem with M·N variables and (M+N) constraints.

The construction of the associated information structure proceeds as described earlier in this section. Each partitioning matrix specifies the components assigned to and processed by the m-th DM. Since the DMs are identical, the particular assignment of partitioning matrices to DMs is arbitrary.

Many Groups of DMs

The assumption that DMs in the g-th group, $g = 1,2,...G$ possess identical properties ($G \geq 2$), does not lead to a reduction in the dimensionality of the GN. The alternative approach of solving M^* KPs cannot be used either, unless the relative efficiency of DMs among groups can be established and ranked, i.e., a DM from M^g can process any set of components more quickly than any DM from $M^{g'}$. If this dominance with respect to processing time functions exists, then the sequential approach used in the previous section could again be used. In particular, if M^g contains the most efficient DMs and d^g is its dimension, then the first d^g KPs would assign components to DMs from this g-th group. DMs from the second most efficient group would be considered next and this procedure would continue until all components were assigned.

Single Group of Identical Components

The third special case illustrates that under the assumptions that

a) all components are independent and

b) all components' alphabets are of the same dimension
    and have identical probability distribution functions,

the problem of selection and assignment of partitions of components to
DMs does not require a GN algorithm or any other mathematical programming
algorithm to solve it. A feasible solution, which requires the minimum
number of DMs, can be obtained directly with relatively few computations.

If the components of a partition vector are mutually independent,
then the entropy associated with that partition is equal to the sum
of entropies of each of the components in the partition, i.e.,

$$H(\underline{z}_{N'}) \equiv H(x_1, \ldots, x_N') = \sum_{n=1}^{N'} H(x_n) = N'H_o$$

The entropy of any partition of dimension s is equal to sH_o.

No restrictions have yet been placed on the DMs as to which
components they may process. Thus, any component(s) that the m-th
DM can process without overload may be assigned to him. Feasibility
with respect to mean processing time requires that

$$t^m(s) + c^m H(\underline{z}_s) = st^m + c(sH_o) \leq \delta, \quad m = 1,2,\ldots,M. \quad (11)$$

The maximum number of components that the m-th DM can process
without overload, s_m^*, is derived from the inequality (11) as follows:

$$st^m + c^m(sH_o) \leq \delta \qquad m = 1,2,\ldots,M$$

$$\Rightarrow s \leq \delta \ (t^m + c^m H_o)^{-1} \qquad m = 1,2,\ldots,M$$

$$\Rightarrow s_m^* = \left\lfloor \delta \ (t^m + c^m H_o)^{-1} \right\rfloor \qquad m = 1,2,\ldots,M$$

where $\lfloor \cdot \rfloor$ denotes the function that yields the greatest integer less than or equal to its argument.

In order to minimize the number of DMs required to process the input vector, components are assigned to those DMs who can process the greatest number of components in time δ. The s_m^*'s are ranked in order of magnitude. The quantity s_{mf}^* is defined to be the f-th largest s_m^*, $m = 1,2,\ldots,M$; $f = 1,2,\ldots,M$. If each of two or more DMs can process the same maximum number of components, then the ranking among them is arbitrary.

To determine M^*, the minimum number of DMs necessary to process the input vector, DMs are added until all components are assigned, i.e.,

$$\sum_{f=1}^{M^*} s_{mf}^* \geq N$$

Note that in this formulation there is no distinction in the cost between DMs of different capabilities.

The DMs corresponding to $f = 1,2,\ldots,M^*$ will be included in the information structure. Each of the first M^*-1 DMs will process partitions of dimension s_{mf}^*. The last DM will process a partition of dimension less than or equal to s_{mM}^*:

$$N - \sum_{f=1}^{M^*-1} s_{mf}^*$$

Since the components all have alphabets with identical probability distributions and DMs are not restricted as to which components they may process, the number of possible assignments is

$$\frac{N! / \prod_{j=1}^{N} h_j}{(s_{m1}^*!)(s_{m2}^*!) \ldots (s_{mM}^*!)}$$

where h_j is the number of groups of dimension j, $j = 1,2,\ldots,N$. Let h_j be unity if no groups of dimension j exist. The generic information structure is:

$$\Pi = (\pi^1_{k_1}, \pi^2_{k_2}, \ldots, \pi^{M^*}_{k_{M^*}})$$

where $\pi^m_{k_f}$ is the k_f-th partitioning matrix of dimension $s^*_{mf} \times N$ and $f = 1,2,\ldots,M^*$; $m = 1,2,\ldots,M^*$.

Many Groups of Identical Components

Let there be G groups of components with components within a group, L^g, $g = 1,2,\ldots,G$, possessing alphabets with identical probability distributions. All components are assumed mutually independent. A component x^g can represent each group L^g.

All possible partitions can be implicitly considered by using only the representative components and the dimension of each group, i.e., G representative components need be considered with each one being assigned s^g times. It is possible that the same representative component may be assigned to the same DM several times. In order to allow for this possibility, several of the constraints must be modified in the general formulation of the problem. In particular, since a representative component x^g can be assigned more than one time, Y_{gm} is no longer restricted to be binary. Rather, it is restricted to be an integer with an upper bound of s^g. The modified constraint set is:

$$Y_{gm} = 0,1,\ldots,s^g \qquad g = 1,2,\ldots,G; \quad m = 1,2,\ldots,M. \qquad (12)$$

To guarantee that every representative component is processed exactly s^g times constraint set (8) is modified to

$$\sum_{m=1}^{M} Y_{gm} = s^g \qquad\qquad g = 1,2,\ldots,G. \qquad\qquad (13)$$

Figure 4 illustrates the reduction of the number of components and, consequently, of the overall dimensionality of the problem. As G, the number of distinct groups, decreases, the dimensionality of the problem is also reduced, since the size of the current problem is a linear function of the number of components.

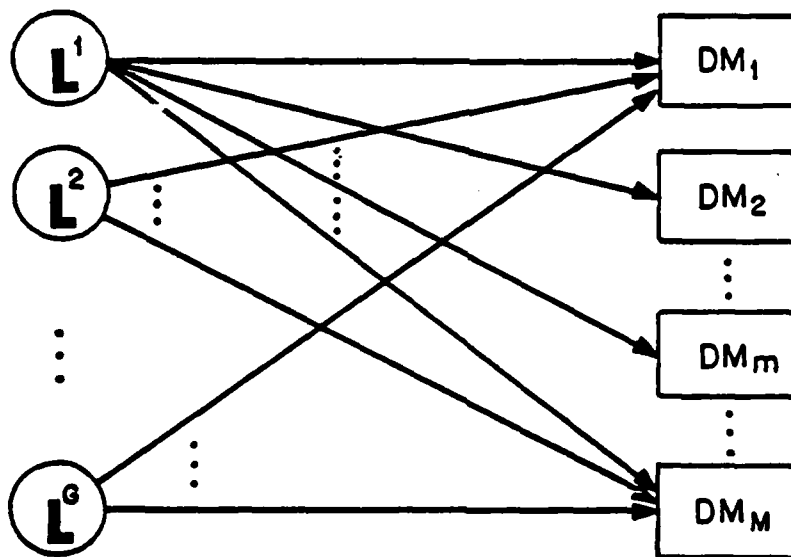## COMPONENTS   DECISION MAKERS

Figure 4. Groups of Identical Components

The information structure can be obtained in a manner similar to that for groups of DMs. Each variable Y which has a nonzero value r^g implies that r^g components from L^g, $g = 1,2,\ldots,G$, are assigned to the m-th DM with the assignment of particular components being arbitrary.

Specialization Among Decision Makers

Specialization restricts the types of components that can be assigned to a particular DM. To imbed this constraint into the GN formulation it is only necessary to restrict the variable Y_{nm}.

If $\underline{x}$ has five components and DM_1 can only be considered for processing components x_1, x_2 and x_3, then variables Y_{11}, Y_{21}, and Y_{31} would be included in the formulation while Y_{41}, and Y_{51} would not be. For each DM let L^m be the set of components that may be assigned to him. Then, the modified formulation is

$$- \sum_{\substack{m=1 \\ n \in L(m)}}^{M} Y_{nm} = -1 \qquad\qquad n = 1, 2, \ldots, N \qquad (14)$$

$$\sum_{n \in L(m)} (t^m + c^m H_n) Y_{nm} \leq \delta \qquad\qquad m = 1, 2, \ldots, M \qquad (15)$$

$$Y_{nm} = 0, 1 \qquad\qquad m = 1, 2, \ldots, M; \; n \in L(m) \qquad (16)$$

The reduction in dimensionality is a function of the size of the sets L^m and the decoupling, which occurs when sets of components can only be assigned to particular sets of DMs.

For example, let $M^1 = (DM_1, DM_2)$, $M^2 = (DM_3, DM_4)$, $L^1 = L^2 = (x_1, x_2, x_3)$ and $L^3 = L^4 = (x_4, x_5)$. Figure 5 illustrates the decoupling of this problem. The specialization restriction reduces the number of variables Y_{nm} that must be considered from 20 to 10. The decoupling effect allows two GNs to be solved, one with 6 variables, the other with 4 variables, rather than one GN with 10 variables. Since the complexity associated with the Branch and Bound algorithm used to solve this problem increases nonlinearly at a rate that is greater than unity, the ability to decouple the problem is a significant benefit.
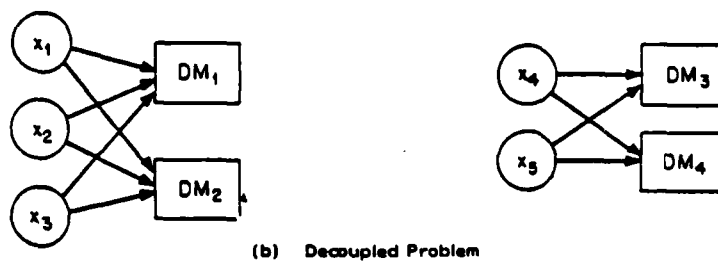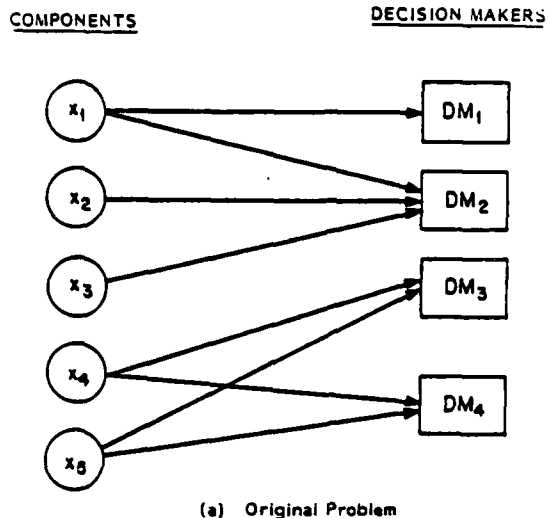
COMPONENTS   DECISION MAKERS

(a) Original Problem

(b) Decoupled Problem

Figure 5. Specialization

Prespecified Grouping of Components

Although components may be mutually independent, reasons may exist
which require certain components to be processed together. To guarantee
these components are all assigned to the same DM, supercomponents are
created. Let the set $L(g)$ contain the set of components comprising this
supercomponent, $H(g)$ be the set's entropy and $s(g)$ be the set's dimension.
The set L of components comprising the input vector is redefined to be
the set of supercomponents $L(g)$ which has dimension G.

The number of components that need be considered explicitly is G.
The only modification to be made to the GN concerns constraint set (10)
which is modified to account for the dimension of the supercomponents:

-21-

$$\sum_{g=1}^{G} (s(g)t^m + c^m H_g) \, Y_{gm} \le \delta \qquad\qquad m = 1, 2, \ldots, M \qquad (17)$$

The solution to the GN formulation is used to construct the information structure. However, the actual components, not the supercomponents, must be included in the information structure. For example, if $Y_{11} = 1$, $Y_{21} = 1$, $Y_{32} = 1$, and $L^1 = (x_1, x_2, x_3)$, $L^2 = (x_4)$, and $N = 4$ then

$$\pi_1^1 = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \pi_2^2 = [0 \quad 0 \quad 0 \quad 1]$$

so that

$$\Pi = (\pi_1^1, \ \pi_2^2)$$

Summary

In this section an implicit approach for allocating data among decision makers for parallel processing has been presented. This approach does not require each possible partition to be considered explicitly. As a result, the size of the problem has been reduced drastically from order 2^N to order N. The resource allocation problem was formulated as a generalized network and solved using one of the efficient GN algorithms. Several cases were examined in some detail under the assumption that all components of the input need not be processed by more than one DM. The weakening of this assumption is discussed in the next section.

REDUNDANCY

Introduction of redundancy in the design of the single echelon requires that the problem be viewed from a different perspective. The concept of distributed database systems (DDSs) more clearly illustrates the need for including redundancy.

Distributed database systems evolved to satisfy several needs. One need was to have the data which required constant updating near the users. A second need was to reduce vulnerability and thus allow the uninterrupted flow of information, should one data center fail.

Redundancy, as interpreted with respect to SE structures, can be integrated with the concept of DDS. The concept of storing processed data requires a reexamination of what a SE structure is. It was initially defined to be a group of DMs who (i) are not hierarchical with respect to the boundary of the organization and (ii) perform a complex task. Two types of tasks now concern the SE. The first (original) task is to process the data of the input vector. The second task is to store the data, be it processed or not. This second task can be incorporated into the original problem by introducing constraints into the system which cause the original definition of efficiency to be violated. The original statement of objective required that a complex task be performed accurately using the minimum number of DMs while satisfying the constraints. Implicit in this objective was that each component of the input vector should be processed exactly once.

Each DM in the DDS processes and stores the entire partition of components he receives. All components are again assumed mutually independent. A subset of these processed components is also transmitted. This subset can range from the entire partition of processed components to the empty set. Time is required to process, store, and transmit components. The storage time and transmitting time functions are

-23-

assumed identical and are imbedded in a single processing time function, τ_k^m. The effect of redundancy on the ability to formulate the problem as a GN as well as on the dimensionality issue is significant. Three cases will be examined.

The first case requires that each component x_n be processed and stored R_n times. If R_n is unity for all n, then this formulation defaults to the original GN formulation, for parallel processing. If $R_n > 1$ for at least one component, then only a minor modification is necessary. Constraint set (8) is modified so that

$$- \sum_m Y_{nm} = -R_n \qquad\qquad n = 1,2,\ldots,N \qquad\qquad (18)$$

The only difference in constructing the information structure is that partitioning matrices and their associated vectors will not necessarily contain distinct components.

The second case requires the R^g DMs must each process a specified set of components. Supercomponents, composed of the specified sets of components, are constructed. The formulation is identical to that for prespecified grouping of components except that the right hand side constants of constraint set (8) are now replaced with R^g. The modified constraint set is

$$- \sum_{m=1}^{M} Y_{gm} = -R^g \qquad\qquad g = 1,2,\ldots,G \qquad\qquad (19)$$

The information structure is constructed in an identical fashion to that for prespecified groupings of components. Again, the associated partitioning matrices and vectors will not necessarily have distinct components.

In the third case, a particular group of components L', must be processed and stored by each member of a prespecified group of DMs, M'. It is assumed that these components do not overload these DMs.

If, however, the time required by the m-th DM to process his assigned components is strictly less than the available time δ, additional components may also be assigned to him provided they do not overload him. An available processing time, δ_m, is defined for each DM in **M'**, which is equal to the difference between the original available processing time and the time required to process the specified components.

Two modifications of the original GN formulation are made:

a) All components whose assignments to DMs have been prespecified are omitted. To guarantee that each of the remaining components is processed once and only once the following constraint set is formulated,

$$- \sum_{m=1}^{M} Y_{nm} = -1 \qquad\qquad n \in L'' \qquad\qquad (20)$$

where L'' is the set of unspecified components.

b) To guarantee that no DM is overloaded the following constraint set is formulated; each δ is replaced with the appropriate δ_m.

$$\sum_{n \in L'} (t^m + c^m H_n) \, Y_{nm} \le \delta_m \qquad m = 1,2,\dots,M \qquad (21)$$

The variables, Y_{nm}, are again restricted to be binary

$$Y_{nm} = 0,1 \qquad\qquad n \in L' \;;\; m = 1,2,\dots,M. \qquad (22)$$

The partitioning matrix associated with each DM of the single echelon (SE) includes all components he was preassigned and all components for which $Y_{nm} = 1$, $n \in L'$. For example, if $N = 3$ and if DM_1 were pre-assigned components (x_1,x_2,x_3) and also were able to process component x_5, so that $Y_{51} = 1$, then his associated partitioning matrix would be:

$$\pi_1^1 = \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{matrix}$$

$$\pi_1^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The information structure is constructed directly.

ALTERNATE PROCESSING

Information structures based on alternate processing are appropriate when the input vector cannot be partitioned, i.e., when the strategy of creating self-contained tasks cannot be used to avoid overload. The other strategy available to the organization designer is the creation of a slack resource, in this case, time. Thus, each DM is given more time to process the input assigned to him, which, as already mentioned, introduces a delay strictly greater than δ .

A deterministic strategy is one in which the ordering of the assignment of the input vectors to the DMs is fixed. In order to specify the optimal information structures associated with this strategy, it is necessary to determine simultaneously:

   a) the minimum number of DMs, M^*, necessary to
       process the input vectors without any one being
       overloaded and

   b) the frequency, q_m, with which each of these DMs
       receives an input vector.

A very simple method for solving this problem exists. The overload constraint requires

$$q_m \leq \delta / \bar{\tau}^m$$

where $\bar{\tau}^m$ is the average time for the m-th DM to process an input vector. Without any loss of generality, the DMs may be re-indexed according to their processing time functions: i.e., let the first DM be the most efficient and the m-th DM be the least efficient, so that $\bar{\tau}^1 \leq \bar{\tau}^2 \leq ... \leq \bar{\tau}^M$. The other constraint on the problem is that all of the data be processed;

$$\sum_{m=1}^{M^*} q_m = 1$$

where M^* has yet to be determined. The solution proceeds by choosing DMs in order of efficiency until

$$\sum_{m=1}^{\lambda+1} \delta/\bar{\tau}^m > 1 \geq \sum_{m=1}^{\lambda} \delta/\bar{\tau}^m \qquad (23)$$

If the right hand side of (23) is an equality, then the minimum number of DMs, M^*, necessary to process the input vectors without overload is equal to λ, and

$$q_m = \begin{cases} \delta/\bar{\tau}^m & \text{if } 1 \leq m \leq \lambda \\ 0 & \text{Otherwise} \end{cases}$$

If the right hand side of (23) is a strict inequality, then the minimum number of DMs, M^*, is equal to $\lambda+1$.

Because $\displaystyle\sum_{m=1}^{\lambda+1} \delta/\bar{\tau}^m > 1$, q_m must be defined as

$$q_m = \begin{cases} \delta/\bar{\tau}^m - \varepsilon^m & \text{if } 1 \leq m \leq \lambda+1 \\ 0 & \text{Otherwise} \end{cases}$$

where $\varepsilon^m \geq 0 \qquad m = 1,2,\ldots,\lambda+1$

$$\sum_{m=1}^{\lambda+1} (\delta/\bar{\tau}^m - \varepsilon^m) = 1$$

These ε^m may be set to ensure that all of the q_m are rational, so that a cyclical strategy can be used.

A cyclical strategy is defined as a strategy in which the ordering of the assignment of the input vectors to the DMs is repeated every δ' input vectors. In the case that the right hand side of (23) is an equality and at least one q_m is irrational, a cyclical strategy cannot be used; but it may be argued that since the $\bar{\tau}^m$'s are usually estimated rather than precisely calculated, they can always be chosen to be rational and so a cyclical strategy may always be used. In this case, δ', the number of inputs in one cycle, is the lowest common denominator of the q_m's. The information structures for a deterministic cyclical strategy may now be completely specified.

Define F to be the ordered set of indices on one cycle of δ' input vectors: that is,

$$F \equiv \{f \mid f = 1, 2, \ldots, \delta'\}$$

Now let F^m be a subset of F where

$$F^m \equiv \{f \in F \mid \text{input } \underline{x}_f \text{ is assigned to DM } m\} \quad m = 1, 2, \ldots, M^*.$$

With the indicator variable ϕ_f^m defined as

$$\phi_f^m = \begin{cases} 1 & \text{if } f \in F^m \quad m = 1, 2, \ldots, M^* \\ 0 & \text{Otherwise} \end{cases}$$

the only requirement on the assignment of input to DM m is that

$$\sum_{f=1}^{\delta'} \frac{\phi_f^m}{\delta'} = q_m \quad m = 1, 2, \ldots, M^*$$

Since input vectors only arrive once every δ time units, $\underline{x}_f$ is assigned at time $t = (k\delta' + f)\delta$, where $k = 0, 1, \ldots$ determines the number of cycles that have been completed at time t. Therefore, the information structure for DM m at time t is given by

$$\pi^m(t) = \pi^m [(k\delta' + f)\delta] = \begin{cases} I & \text{if } f\epsilon F^m \quad m = 1,2,\ldots,M^* \\ 0 & \text{Otherwise} \end{cases} \qquad (24)$$

A special case of the deterministic cyclical strategy is a periodic one characterized by the following properties:

a) the length of the sequence is exactly M^*, the number of decision makers, and

b) each DM is assigned exactly one symbol during the execution of the sequence.

The information structure for the periodic strategy is given by

$$\pi^m(t) = \begin{cases} I & t = (kM^* + m)\delta \\ 0 & \text{Otherwise} \end{cases} \quad m = 1,2,\ldots,M^* ; \quad k = 0,1,2,\ldots \qquad (25)$$

The relative frequency for a DM receiving a symbol for processing in the periodic case is

$$q_m = 1/M^* \qquad \text{for all } m \qquad (26)$$

In order that no DM be overloaded, the mean symbol processing time must satisfy the following inequality:

$$\bar{\tau}^m \leq M^* \delta \qquad m = 1,2,\ldots,M^*$$

In this section, special cases are again explored, as in the section on parallel processing, to provide insight into the effect various properties have on the design problem.

Identical Decision Makers

In the first case, all DMs are assumed to possess identical properties:

a) the processing time functions of all the DMs are
identical,

b) the DMs are able to process any input symbol, and

c) they have identical costs.

Assumption (a) implies that only one mean processing time, $\bar{\tau}$, need be
computed. Since the DMs are identical it follows that

$$q_m = q \leq \delta / \bar{\tau} \qquad m = 1, 2, \ldots, M. \qquad (27)$$

The only other constraint requires

$$\sum_{m=1}^{M} q_m = M^* q = 1 \qquad (28)$$

Therefore, M^* is the smallest integer that satisfies eq. (28) subject
to the condition (27). The resulting strategy is periodic: the solution
only requires that the relative frequency of symbol processing by a DM
be $1/M^*$.

The corresponding information structure is

$$\pi^m(t) = \begin{cases} I & \text{for } t = (kM^* + m)\,\delta \\ & \\ 0 & \text{Otherwise} \end{cases} \qquad m = 1, 2, \ldots, M^*; \ k = 0, 1 \ldots$$

Groups of Identical Decision Makers

Let there be G groups of decision makers denoted by M^g, $g = 1, 2, \ldots, G$
with the DMs within each group possessing identical properties. Let d_g be
the dimension of M^g and let $\bar{\tau}^g$ be the mean processing time for each member
of M^g. As in the introductory section on alternate processing, the groups
are re-indexed according to their processing time functions so that the

DMs in $\mathbf{M}^1$ are the most efficient and those in $\mathbf{M}^G$ are the least efficient: i.e., $\bar{\tau}^1 \leq \bar{\tau}^2 \leq \ldots \leq \bar{\tau}^G$. Then groups of DMs are chosen in order of efficiency until

$$\sum_{g=1}^{\lambda+1} d_g \left(\frac{\delta}{\bar{\tau}^g}\right) > 1 \geq \sum_{g=1}^{\lambda} d_g \left(\frac{\delta}{\bar{\tau}^g}\right) \tag{29}$$

If the right-hand side of (29) is an equality, then

$$M^* = \sum_{g=1}^{\lambda} d_g. \qquad \text{and}$$

$$q_m = \begin{cases} \delta / \bar{\tau}^g & m \in \mathbf{M}^g; \ g = 1,2,\ldots,\lambda \\ \\ 0 & \text{Otherwise} \end{cases}$$

If the right-hand side of (29) is an inequality, then all of the DMs in $\mathbf{M}^{(\lambda+1)}$ may not be needed. To calculate the minimum number of DMs from $\mathbf{M}^{(\lambda+1)}$, λ^*, necessary to add to the λ groups of DMs already selected, a procedure identical to that described in the introductory section on alternate processing is employed. Then

$$M^* = \left(\sum_{g=1}^{\lambda} d_g\right) + \lambda^*$$

$$q_m = \begin{cases} \dfrac{\delta}{\bar{\tau}^g} - \varepsilon^m & \begin{array}{l} m \in \mathbf{M}^g \ g = 1,2,\ldots,\lambda \text{ or} \\ m \in \widetilde{\mathbf{M}}^{\lambda+1} \subset \mathbf{M}^{\lambda+1} \text{ where the} \\ \text{dimension of } \widetilde{\mathbf{M}}^{\lambda+1} = \lambda^* \end{array} \\ \\ 0 & \text{Otherwise} \end{cases}$$

$$\varepsilon^m \geq 0$$

$$\sum_{m=1}^{M} q_m = 1$$

The information structures for both of these cases may be defined exactly as in (24).

The alternate processing mode has been introduced as a means of implementing the second information reduction strategy, i.e., creation of a slack resource (time) to reduce information overload, as discussed in the introduction to this paper.

INFORMATION STRUCTURES FOR SINGLE ECHELONS

In the previous three sections a detailed analysis of pure parallel
and alternate information structures for the single echelon was presented.
These two basic structures were explored under a variety of assumptions
for the input symbols and the decision makers. The emphasis in the
development was on procedures for allocating the input symbols and
for determining the minimum number of decision makers needed to im-
plement a specific structure. In the general case, however, the or-
ganization designer is given only the properties of the symbol source
and a limited number of decision makers who could form the single echelon.
What is needed is a methodology for the design of the information struc-
ture. Such a methodology, based on the results obtained so far is pre-
sented in this section. It is then applied to an illustrative problem
that requires integration of both parallel and alternate processing
modes.

The first two steps have already been presented. In the first one,
the task is modeled as a source that generates vector signals for the
organization to process. The second step consists of modeling the
decision makers according to their processing time functions and their
specialization.

STEP 3: <u>Information Reduction Strategy</u>

The two basic strategies are

    a)   creation of self-contained subtasks, and

    b)   creation of slack resources.

The first strategy is applicable when the input vector can be partitioned
into subvectors. The second strategy is feasible when the organization
can tolerate some delay (beyond the mean interarrival time $\bar{s}$) in the

processing of the input vector. It is here that the organization designer's understanding of the task to be performed by the organization is crucial. He has to determine the extent to which parallel processing can be used and estimate the maximum tolerable delay. The latter determines the extent to which alternate processing can be used. Assuming that the overall task can be accomplished by various combinations of alternate and parallel processing, he then has to assess the trade-offs between subdivision into smaller independent tasks and longer delays.

Once he selects an integrated information reduction strategy, the designer proceeds to Steps 4 to 6. When he evaluates the resulting design, he can return to Step 3 and modify the strategy.

STEP 4: <u>Mathematical Model</u>

The next step consists of the formulation of the mathematical model that represents the integrated *information reduction strategy* selected in Step 3. There are four basic constraints, common to all strategies considered so far, that must be expressed analytically.

a) All components must be processed. This is a key assumption; if any data were to be rejected, then their sources have to be eliminated from the super-source model.

b) No decision maker is overloaded. This condition requires that in the final design the mean processing time of each decision maker in the echelon does not exceed the mean interarrival time for symbols (or tasks) <u>received by him.</u>

c) Only decision makers who receive data from the supersource are members of the single echelon. The decision makers who are not used in the single echelon are assigned the null or empty subtask.

d) Each DM is assigned at most one subtask. Additional
    constraints that are specific to the particular appli-
    cation can be introduced.

STEP 5: Optimization Problem

Each DM is assigned a cost that may depend on his capabilities
and limitations in performing the tasks. Then the objective function
to be minimized is the total cost of the DMs included in the single
echelon. If the costs associated with each DM are assumed equal, then
the optimization problem reduces to one of minimizing the number of
members in the echelon.

In various cases, under particular simplifying assumptions, solutions
were obtained from reasonably simple and straightforward computations.
If this was not the case, mathematical programming proved an attractive
method for obtaining solutions. In particular, generalized network (GN)
formulations proved most attractive because of the efficient algorithms
which exist to solve them. Knapsack problems, and mixed integer linear
programs can also be used.

SETP 6: Information Structures

The solution to the optimization problem yields the number (and
identity) of the DMs comprising the single echelon and the assignment
of a task or subtask to each DM. The results are expressed formally
in terms of an information structure that consists of parallel or
alternate processing, or a combination of both. The structure is
evaluated to determine whether the tradeoffs between the number of
DMs and delays are acceptable; if not, then the designer should return
to Step 3 and revise the information reduction strategy.

The six steps are applied now to a design problem that illustrates
the methodology.

Problem

Consider G distinct sources, each source generating a vector of signals. The task is such that each source output has to be processed intact, i.e., it cannot be partitioned. There are M decision makers who can receive the generated signals and none of these DMs can process the output of any of the G sources without being overloaded. A parallel/alternate information structure seems appropriate.

STEP 1: Tasks

The supersource consists of G synchronized sources that generate vector signals. The mean signal generation rate is δ^{-1}. The elements of the input vector can be partitioned into G sets, each set corresponding to the output of each of the individual sources. This is the finest grain decomposition of the input.

STEP 2: Decision Makers

The DMs constitute a group of M distinct members.

STEP 3: Information Reduction Strategy

The decomposition of the input vector allows for the parallel processing of the signals generated by the G sources. No further division into subtasks is possible. Since every one of the G subtasks arriving at a rate δ^{-1} cannot be processed by any DM without causing overload, the second information reduction strategy (creation of slack resources) must be used. Alternate processing of signals generated by each source would allow additional time for each DM to do the processing and therefore, overload may be avoided. The resulting processing mode is an integrated parallel/alternate processing.

-37-

STEP 4: <u>Mathematical Model</u>

a) Since alternate processing is assumed for the output
of each source, the requirements that all signals be
processed reduces to the condition that the sum of
the symbol assignment frequencies for the output of
each source, q_{gm}, must be equal to unity. This is
expressed as eq. (30)

$$\sum_{m=1}^{M} q_{gm} = 1 \qquad g = 1,2,\ldots,G \qquad (30)$$

b) In order that no DM be overloaded, the frequency with
which each receives a signal for processing should be
sufficiently low so that his mean processing time
does not exceed the effective mean interarrival time.
This condition is given by eq. (31)

$$0 \leq q_{gm} \leq \delta / \bar{\tau}_g^m \qquad g = 1,2,\ldots,G; \quad m = 1,2,\ldots,M \qquad (31)$$

c,d) Any DMs that receive input for processing with zero
frequency are excluded from the single echelon. Further-
more each DM is allowed to receive inputs from at most
one of the G sources. Constraints (32) and (33) guarantee
these conditions where the binary variable Y_{gm} is <u>zero</u>
when the m-th decision maker is assigned the output of
the g-th source.

$$\sum_{g=1}^{G} Y_{gm} = G-1 \qquad m = 1,2,\ldots,M \qquad (32)$$

$$Y_{gm} \cdot q_{gm} = 0 \qquad g = 1,2,\ldots,G; \quad m = 1,2,\ldots,M \qquad (33)$$

$$Y_{gm} = 0,1 \qquad g = 1,2,\ldots,G; \quad m = 1,2,\ldots,M \qquad (34)$$

STEP 5: Optimization Problem

In a structure such as this, the number of decision makers that can process the incoming signals is a reasonable objective function to be minimized. Note that since the DMs do not have identical properties, the problem cannot be decoupled into G distinct optimization problems even though no decision maker is allowed to process signals from more than one source. The resulting mathematical programming problem is difficult to solve because of the nonlinearity of the constraint (33).

STEP 6: Information Structures

The information structure can be obtained directly from the solution to the optimization problem (the nonlinear MP). The single echelon is composed of only DMs for which the corresponding frequency q_{gm} is strictly positive. The information structure, Figure 6, specifies the decision maker, the group of components g he processes and the frequency with which he is assigned these inputs. It is given by

$$\pi_g^m(t) = \pi_g^m \left[(k \, \delta'_g + f) \, \delta \right] = \begin{cases} 1 & \text{if } f \in F_g^m \; ; \; k = 0,1,\ldots \\ \\ 0 & \text{Otherwise} \end{cases} \qquad (35)$$
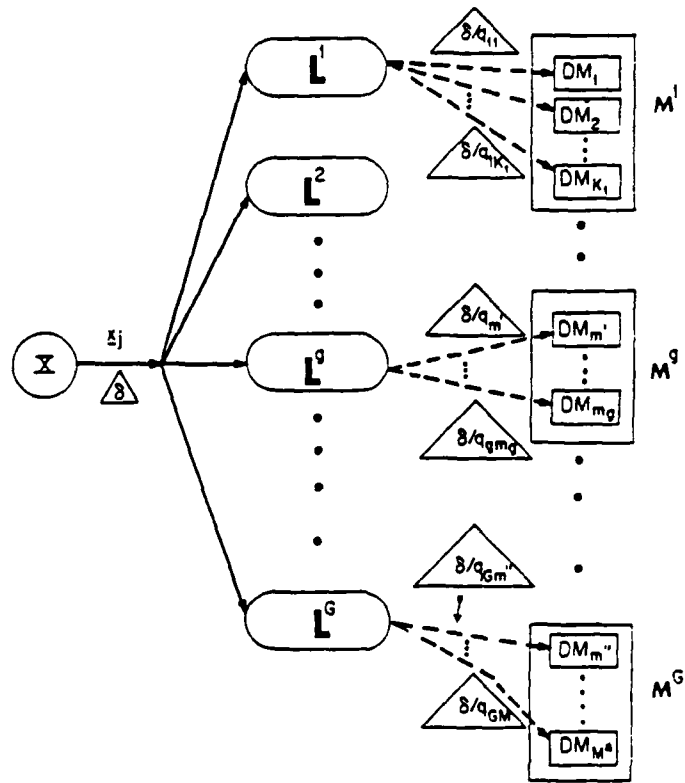
where δ'_g is the lowest common denominator of the q_{gm}, $m = 1,2,\ldots,M$ and

$$F_g^m \equiv \{f \mid \text{input } \underline{x}_f \text{ from source g is assigned to m-th DM}\}$$

$$m = 1,2,\ldots,M \; ; \quad g = 1,2,\ldots,G$$

where $\triangle{8}$: input symbol generation time

Figure 6.   Parallel/Alternate Processing

Note that while the inputs from the sources are received by the single echelon simultaneously, the outputs are not synchronized.   Indeed, each DM introduces a different delay; the maximum delay is given by the maximum value over m of

$$\delta \, (1-q_{gm}) \, / \, q_{gm}$$

If this delay is unacceptable, then more efficient DMs are needed.

The solution to the problem is illustrated now with a specific example.

Let there be three distinct sources (G = 3) each generating a vector $\underline{x}_g$ in synchronization and let the rate be $\delta^{-1} = 1$.

$$L^1 : \{x_1, x_2, x_3\}$$
$$L^2 : \{x_4\}$$
$$L^3 : \{x_5, x_6\}$$

Let the size of the alphabets of the elements of L^1 be ten, of L^2 four, and of L^3 eight. Assume that the corresponding probability distributions are uniform. The M decision makers are assumed identical with processing time functions given by

$$\tau_g^m = 0.5 + 0.25 \sum_{x_i \in L^g} H(x_i) \qquad \text{for all } m, g.$$

The processing times for each group g are computed readily:

$$\tau_1^m = 0.5 + 0.25 \times 3 \times \log_2 10 = 2.99$$

$$\tau_2^m \quad 0.5 + 0.25 \times 1 \times \log_2 4 = 1.0$$

$$\tau_3^m \quad 0.5 + 0.25 \times 2 \times \log_2 8 = 2.0$$

Inequality (31) yields

$$q_{1m} \leq 1/2.99$$
$$q_{2m} \leq 1$$
$$q_{3m} \leq 1/2$$

Application of (30) leads to:

$$q_{11} = q_{12} = q_{13} = 1/3$$
$$q_{24} = 1$$
$$q_{35} = q_{36} = 1/2$$

with all other q_{gm} equal to zero. Constraints (32), (34) are satisfied by choosing Y_{gm} equal to zero when $q_{gm} \neq 0$ and equal to unity when $q_{gm} = 0$.

To construct the information structure in accordance with (35), several quantities must be defined. Clearly,

$$\delta_1 = 3 \; ; \; \delta'_2 = 1 \; ; \; \delta'_3 = 2$$

since a periodic alternating strategy is appropriate. Also

$$F_1 = \{1,2,3\} \text{ with } F^1_1 = \{1\} \; ; \; F^2_1 = \{2\} \; ; \; F^3_1 = \{3\} \; ;$$

$$F_2 = \{1\} \qquad \text{with } F^4_1 = \{1\} \; ;$$

$$F_3 = \{1,2\} \quad \text{with } F^5_3 = \{1\} \; ; \; F^6_3 = \{2\} \; .$$

Then, for $k = 0,1,2,\ldots$, the partition matrices are given by

$$\pi^m_1 (t) = \pi^m_1 (3k + f) = \begin{cases} I & \text{if } f \in F^m_1 \\ \\ 0 & \text{Otherwise} \end{cases}$$

$$\pi^m_2 (t) = \pi^4_2 (k + 1) = I$$

$$\pi^m_3 (t) = \pi^m_3 (2k + f) = \begin{cases} I & \text{if } f \in F^m_3 \\ \\ 0 & \text{Otherwise} \end{cases}$$

and, finally, the information structure can be constructed:

$$\Pi(t) = \left(\pi^1_1(t), \; \pi^2_1(t), \; \pi^3_1(t), \; \pi^4_2(t), \; \pi^5_4(t), \; \pi^6_3(t) \right) .$$

The minimum number of DMs is six and the maximum delay is three units (i.e., 3δ's). The $\Pi(t)$ specifies completely the allocation of input signals to each member of the single echelon.

CONCLUSION

An approach to the design of information structures for a single
echelon organization has been presented. This approach is based on
the properties of the inputs, the characteristics of the available
decision makers, and the constraints imposed on the organization by
the task. Two basic information reduction strategies, creation of
self-contained tasks and creation of slack resources, were modeled
as parallel and alternate processing, respectively. It was then
shown that complex information structures can be constructed using
combinations of parallel and alternate processing. The former is
appropriate when an overall task can be divided into subtasks; the
latter, when delays in producing an output can be tolerated and
the task cannot be divided.

The next major step in this research is the integration of the
single echelon with other parts of the organization. The single
echelon is responsible for transmitting the processed inputs to
the appropriate destinations within the organization. This trans-
mission of processed data to other members in the organization is
referred to as serial processing.

The design of multiechelon structures requires each echelon to
process its information without overload. The constraints on each
echelon, however, must be inferred from the constraints that are
imposed on the overall organization. This introduces a higher
level of complexity to the design problem.

REFERENCES

[1] E. R. Lawler III and J. G. Rhode, <u>Information and Control in Organizations</u>, Goodyear Publishing Company, Inc., Pacific Palisades, CA. 1976.

[2] J. Galbraith, <u>Designing Complex Organizations</u>, Addison-Wesley, New York, 1973.

[3] T. B. Sheridan and W. R. Ferrell, <u>Man-Machine Systems</u>, The M.I.T. Press, Cambridge, MA. 1974.

[4] R. F. Drenick, "Organization and Control," in <u>Directions in Large Scale Systems</u>, Y. C. Ho and S. K. Mitter, Eds. Plenum Press, N.Y. 1976.

[5] R. V. Hogg and A. T. Craig, <u>Introduction to Mathematical Statistics</u>, MacMillan, New York, 1968.

[6] S. P. Bradley, A. C. Hax and T. L. Magnanti, <u>Applied Mathematical Programming</u>, Addison-Wesley, Reading, MA. 1977.

DATE FILMED

FILMED

4-8