Evaluation of Purposive Organizational Networks: Some Computational Methods

Masao Tsuji

I. Introduction[1]

For a business corporation, a not-for-profit organization, an administrative agency, the usual hierarchic chart ("who reports to whom?") *should be replaced by a more specific network of tasks*: "who performs which action, or learns what, or tells what to whom, in response to what information?" Each member receives, as inputs, messages from other members or from the outside. He transforms them into his outputs: messages to other members or to the outside world. Each member's task is thus a transformer, or "processor". The messages from and to the outside world are, respectively, the "events" (e. g., next month's market prices and wages) and the members' "decisions" (e. g., the production volumes for various products of the firm). Together, events and decisions determine the "outcome". For example, one member estimates future market conditions, or the arrival times of various supplies, etc. He communicates the estimates, or its different aspects or details, to several other members, depending on their respective special tasks. Some of them, on the basis of this and other messages received, may decide on "final actions" within their respective competences and / or decide on what to communicate to whom.

Given the actual external conditions (as distinct from their esti-

(1) A more complete description of the underlying rationale was given by Jacob Marschak "Efficient Organizational Design" in *Economic Theory for Economic Efficiency: Essays in Honor of Abba P. Lerner* (in press). Also issued as Working Paper No. 273, Western Management Science Institute, UCLA.

mates), the organization's *benefit* will depend on the members' final actions and therefore on the manner in which the message-processing tasks were defined and assigned. This assignment will also determine the ("managerial") costs to the organization.

In general, a processor's output message in response to the input is uncertain, i. e., "noisy". The processor is represented by a "transition matrix" of output probabilities (1's and 0's in the special, "noiseless" case), given each possible input. The network, coupling specified processors in specified ways, results, then, in an "overall transition matrix" of probabilities of decisions, given the events.

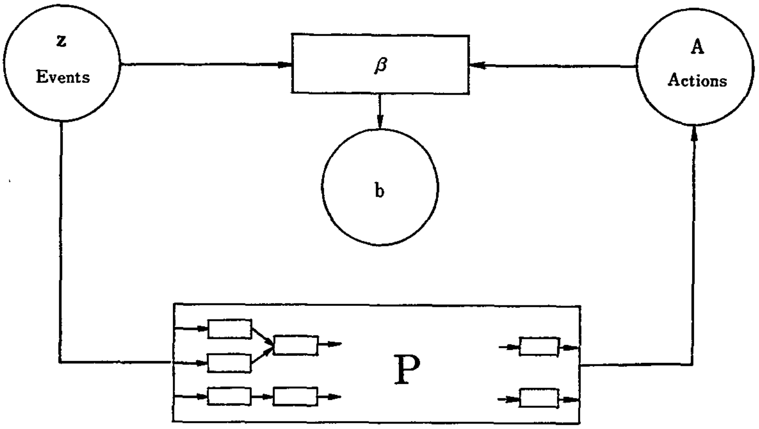

Fig. 1. 1 Organizational Decision-Making

The overall transition matrix is represented by a large box, denoted by **P** in Fig. 1. 1. Given events $z \in Z$, actions $a \in A$, benefit function β, and probability densities $\pi(z)$, the expected gross benefit b may be calculated as follows:

(1. 1) $b \equiv B(\mathbf{P}; \beta, \pi) = \sum_{a,z} \beta(a,z)\ \pi(z)\ \mathbf{P}(a|z)$

A purposive organization can be regarded as being designed by an "organizer" (e. g., a management consultant, or the head of a firm) who prefers networks with higher "expected utility to the organization" as visualized by him. For simplicity, this is approximated by the expected difference between the "benefit" of the outcome of events and decisions, and the "managerial costs" of task performance. This expected difference is determined by the network's overall transition matrix of events into decisions, given the organizer's estimate of event probabilities of the organizational benefit function of events and decisions and of the costs of component processings.

The purpose of this paper is to develop methods to compute the overall transition matrix implied by a given network of tasks in an organization. It will be shown that to compute the overall transition matrix, both the conventional and the direct (=Kronecker) multiplication of component matrices for coupling "in series" or "in parallel", respectively, are not sufficient. When two members receive, two *related messages* (e. g., identical, partly overlapping, or one a condensed version of the other), matrix multiplication must be appropriately modified.

II. Coupling in Series: The Conventional Products of Matrices

In an organization each member receives, as inputs, messages from other members or from the outside world. He processes them into his outputs: messages to other members or decisions. Each member's task is thus called "processor".

Definition 2. 1. Processor P_i represents member i's task which transforms an input message $x \in X \equiv \{a_1, a_2, \cdots\cdots a_m\}$ into an output message $x' \in X' \equiv \{b_1, b_2, \cdots\cdots b_{m'}\}$ and is characterized by a Markov matrix with m rows and m′ columns,

$$(2.\ 1) \qquad P_i \equiv \begin{pmatrix} p_{11} & p_{12} \cdots\cdots & p_{1m'} \\ p_{21} & p_{22} \cdots\cdots & p_{2m'} \\ & p_{ij} & \\ p_{m1} & p_{m2} \cdots\cdots & p_{mm'} \end{pmatrix}$$

where p_{ij} is the probability that if member i receives input message a_i, he sends output message b_j.

It is convenient to label the input and output message spaces by the row indeces $\{a_1, a_2, \cdots\cdots, a_m\}$ and the column indeces $\{b_1, b_2, \cdots\cdots, b_{m'}\}$. These probabilities, p_{ij}, must satisfy

$$0 \leq p_{ij} \leq 1 \quad \text{and} \quad \sum_{j=1}^{m'} p_{ij} = 1.$$

Definition 2. 2. Processors P_i and P_{i+1} are said to be coupled in series iff $x'_i = x_{i+1}$. In other words, the output message x'_i from Processor P_i is the only input message to Processor P_{i+1}.

Definition 2. 3. Processors $P_1, P_2, \cdots\cdots, P_n$ are said to be a stochastic chain if $x'_i = x_{i+1}$ for $i = 1, 2, \cdots\cdots, n$, as shown in Fig. 2. 1.

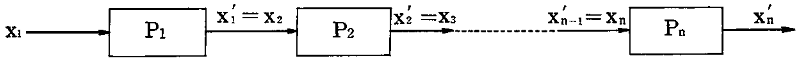

Fig. 2. 1 Stochastic Chain

Note that a stochastic chain defined above has the Markovian property in the sense that for $k = 0, 1, 2, \cdots\cdots$ and every sequence $a_i, b_j, k_1, k_2, \cdots\cdots, k_n$,

$$(2.\ 2) \quad p(x_{k+1}=b_j \mid x_1=a_{k_1}, x_2=a_{k_2}, \cdots\cdots, x_{n-1}=a_{k_{n-1}}, x_k = a_j) = p(x_{k+1}=b_j \mid x_k=a_i)$$

However, the chain does not posses stationary transition probabilities since $p(x_{k+1}=b_j \mid x_k=a_i)$ is not independent of k.

Theorem 2. 1. Let $P_1, P_2, \cdots\cdots, P_n$ be stochastic processors coupled in series, then the chain can be represented by a conventional matrix product,

$$(2.\ 3) \qquad P_{1n} = P_1 \cdot P_2 \cdots\cdots P_n$$

where the dot represents conventional matrix multiplication. P_{1n} is a Markov matrix which represents the probabilities of x'_n given x_1.

Proof. It is obvious that the result is true for $n = 1$. For $n = 2$,

$$P_{12} \equiv [p(x'_2 \mid x_1)] = [\sum_{x'_1} p(x'_2, x'_1 \mid x_1)]$$
$$= [\sum_{x'_1} p(x'_1 \mid x_1) \cdot p(x'_2 \mid x_1, x'_1)]$$

Since $\quad p(x'_2 \mid x'_1, x'_1) = p(x'_2 \mid x_1, x_2) = p(x'_2 \mid x_2)$, we can write

$$P_{12} = [\sum_{x'_1} p(x'_1 \mid x_1) \cdot p(x'_2 \mid x_2)] = P_1 \cdot P_2.$$

To establish the truth of the result for n = k + 1, assuming that it is true for n = k, we should apply the same procedure as for n = 2. That is, P_{1k} is multiplied by P_{k+1} to obtain P_{1k+1}. This completes the inductive proof.

III. Coupling in Parallel with Unrelated Inputs: Kronecher Product

When we deal with the case in which there are more than two input sets, the relation between these sets must be examined.

Definition 3. 1. Two finite random sets $X = \{x\}$ and $Y = \{y\}$ are called *unrelated* if the joint probability $p(x, y) > 0$ for all pairs $(x, y) \in X \times Y$. Otherwise, X and Y are called *related.*

Theorem 3. 1. If for all x and y, $p(x, y) = p(x) \cdot p(y)$ (statistical independence of X and Y), then X and Y are *unrelated.* But the converse is not true.

Proof. Sufficiency: Since $p(x) > 0$ and $p(y) > 0$, then $p(x) \cdot p(y) > 0$ for all pairs (x, y). No necessity: If X is statistically dependent of Y, then $p(x, y) = p(x|y) \cdot p(y) \neq p(x) \cdot p(y)$.

Definition 3. 2. Two Markov matrices $P \equiv [p(x'|x)]$ and $Q \equiv [p(y'|y)]$ are said to be coupled in parallel if $p(x'|x)$ and $p(y'|y)$ are uniquely determined by x and y, respectively (See Fig. 3. 1.)

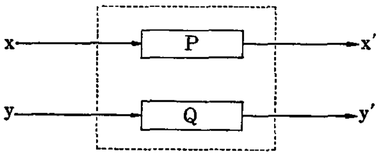

Fig. 3. 1 Coupling in Parallel

We now investigate how to combine two unrelated Markov matrices coupled in parallel into a single matrix, as illustrated in Fig. 3. 1. The combined matrix represents the conditional probabilities of outputs x′ and y′ given inputs x and y. Our effort will be simplified if we first introduce the concept and some of the properties of the Kronecker productof matrices.[2]

Definition 3. 3. Let $A=[a_{ij}]$ be an m by m′ matrix and B be an n by n′ matrix. Then the mn by m′ n′ matrix

(3. 1) $$C \doteq [a_{ij}\ B] = A \times B$$

is called the *Kronecker product or direct product* of A and B.

The matrix C is computed in such a way that each component of A is multiplied by the whole of B. For example if

$$A=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \text{and} \quad B=\begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix}$$

we would have

(3. 2) $$C=A\times B=\begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix}$$

$$=\begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{11}b_{13} & a_{12}b_{11} & a_{12}b_{12} & a_{12}b_{13} \\ a_{11}b_{21} & a_{11}b_{22} & a_{11}b_{23} & a_{12}b_{21} & a_{12}b_{22} & a_{12}b_{23} \\ a_{21}b_{11} & a_{21}b_{12} & a_{21}b_{13} & a_{22}b_{11} & a_{22}b_{12} & a_{22}b_{13} \\ a_{21}b_{21} & a_{21}b_{22} & a_{21}b_{23} & a_{22}b_{21} & a_{22}b_{22} & a_{22}b_{23} \end{pmatrix}$$

The Kronecker product has many of the properties which justify the name "product".

(i) The relation is associative:

(3. 3) $A\times(B\times C)=(A\times B)\times C$

(ii) It is distributive:

(3. 4) $(A+B)\times(C+D)=A\times C+A\times D+B\times C+B\times D$

(3. 5) $(A\times B)\times(C\times D)=(A\ C)\times(B\ D)$

(iii) It is true that

(3. 6) $A^{k+1}=A\times A^{k}$

(3. 7) $(A\times B)^{k}=A^{k}\times B^{k}$

As we will see, the Kronecker product provides a way to combine two processors with unrelated inputs and to yield the desired Markov matrix.

(2) I am indebted to Professor Rudolph Drenick for bringing this method to my attention.

Theorem 3. 2. If two processors P and Q whose input sets x and y are unrelated are coupled in parallel (as illustrated in Fig. 3. 1), then

(3. 8) $$R \equiv [p(x', y'|x, y)] = P \times Q$$

Proof. Since P and Q are coupled in parallel, we have

$$p(x'|x, y, y') = p(x'|x)$$
$$p(y'|y, x, x') = p(y'|y).$$

It then follows that

$$p(x', y'|x, y) = \frac{p(x', y', x, y)}{p(x, y)} = \frac{p(x'|y', x, y)\ p(y', x, y)}{p(x, y)}$$
$$= \frac{p(x'|x)\ p(y'|x, y)\ p(x, y)}{p(x, y)}$$
$$= p(x'|x)\ p(y'|y).$$

Theorem 3. 3. A Kronecker product of two Markov matrices is also a Markov matrix.

Proof. Suppose that $P=[p_{ij}]$ and $Q=[q_{hk}]$ are m by m′ and n by n′ Markov matrices. The Kronecker product R is given by

$$R \equiv [r_{st}] = P \times Q = \begin{pmatrix} p_{11}Q & p_{12}Q & \cdots\cdots p_{1m'}Q \\ \vdots & \vdots & \vdots \\ p_{i1}Q & p_{i2}Q & \cdots\cdots p_{im'}Q \\ \vdots & \vdots & \vdots \\ p_{m1}Q & p_{m2}Q & \cdots\cdots p_{mm'}Q \end{pmatrix}$$

By definition, $\sum_{j=1}^{m'} p_{ij} = 1$ and $\sum_{k=1}^{n'} q_{hk} = 1.$

It will be shown that the sum of any row in R is equal to a unity. For an arbitrary row of R,

$$\sum_{t=1}^{m'n'} r_{st} = p_{i1}\sum_{k=1}^{n'} q_{hk} + p_{i2}\sum_{k=1}^{n'} q_{hk} + \cdots\cdots + p_{im'}\sum_{k=1}^{n'} q_{hk}$$
$$= \left(\sum_{=1}^{m'} p_{ij}\right)\left(\sum_{k=1}^{n'} q_{hk}\right)$$
$$= 1$$

It is important to note that *Theorem 3. 2* and *Theorem 3. 3* is valid for those and only those pairs (x, y) whose joint probabilities are not zero. Suppose that $Z \in X \times Y$ consists of all pairs (x, y) for which $p(x, y)=0$. The Kronecker product R contains as its rows all combinations of input messages x and y. If $Z \neq \phi$, the rows of R include irrelevant pairs which correspond to elements of Z.

IV. Cases of Insufficiency of Conventional and Kronecker Products

In this section we show that the conventional and the direct (= Kronecker) multiplication of matrices are not sufficient to compute the overall transition matrix. When two members receive and process in parallel two related messages, the direct matrix multiplication must be appropriately modified.

Consider the following three cases in which two input messages are related when two processors are coupled in parallel:

(i) two members receive identical messages;

(ii) one member receives the message which is functionally related to the other message, for example, one a condensed version of the other; and

(iii) two members receive partly overlapping messages.

First, suppose that two processors, characterized by Markov matrices P and Q, are connected in parallel as shown by Fig. 4. 1. Processors P and Q receive an identical input message, denoted by x which ranges over a finite set of elements $\{a_1, a_2, \cdots\cdots, a_m\}$. P transforms x into an output $x' \in X' \equiv \{c_1, c_2, \cdots\cdots, c_{m'}\}$ while Q produces an output $y' \in Y' \equiv \{d_1, d_2, \cdots\cdots, d_{n'}\}$.

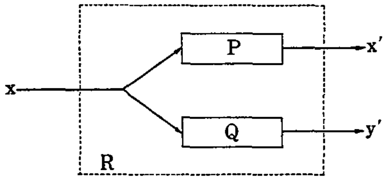

Fig. 4. 1 Two Processors in Parallel: Identical Input

For convenience' sake, we write P and Q as follows:

$$P \equiv [p(x'|x)] \equiv \begin{bmatrix} p(x'|x=a_1) \\ p(x'|x=a_2) \\ \vdots \\ p(x'|x=a_m) \end{bmatrix}$$

(4. 1)

$$Q \equiv [p(y'|x)] \equiv \begin{bmatrix} p(y'|x=a_1) \\ p(y'|x=a_2) \\ \vdots \\ p(y'|x=a_m) \end{bmatrix}$$

Recall that our problem is: Given P and Q, to compute the matrix R of conditional probabilities of x′ and y′ given x:

(4. 2) $$R \equiv p(x', y'|x) \equiv \begin{bmatrix} p(x', y'|x=a_1) \\ p(x', y'|x=a_2) \\ \vdots \\ p(x', y'|x=a_m) \end{bmatrix}$$

By definition, to compute the direct product $P \times Q$ each component of P is multiplied by the whole of Q so that it contains irrelevant rows such as $p(x', y'|x=a_i, x=a_j)$. Obviously, direct multiplication of $[p(x'|x=a_i)]$ and $[p(y'|x=a_j)]$ for $i \neq j$ is not permissible.

Secondly, it may happen that two input messages x and y which are recognized as different sets are functionally related, i. e., $y=f(x)$. For example, x represents detailed description of events while y summarized data of x. Then, we cannot define the joint conditional probabilities of x′ and y′ given x and $y \neq f(x)$ since the marginal joint prodadility $p[x=a_i, y \neq f(a_i)]=0$ for any a_i.

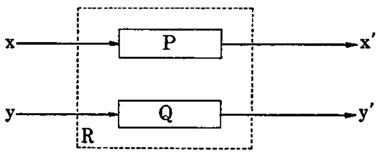

Fig. 4. 2 Two Processors in Parallel: Functional Relation

Fig. 4. 2 is identical with Fig. 3. 2 except for the fact that y is a function of x.

The third is the simplest case among what we call "mixed coupling," which is illustrated in Fig. 4. 3.[3]

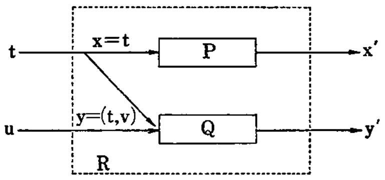

Fig. 4. 3 Mixed Coupling

Let x be an m-tuple, $x=(x_1, \cdots\cdots, x_m)$ and y be an n-tuple, $y=(y_1, \cdots\cdots, y_n)$. Then P and Q are said to be in mixed coupling if and only if for some i and j ($i=1, \cdots\cdots, m$, and $j=1, \cdots\cdots, n$), $x_i=y_j$. In a business organization we may interpret P as a sales manager who receives market information t and Q as a production manager who needs inventory information v as well as t in deciding about production quantity. It is not permissible for a market forecaster to send contradictory messages to P and Q. That is, it should not happen that P receives a message $t = a$ while Q receives a message $t \neq a$. Like the above two cases, however, the direct product of P and Q inevitably involves such multiplication as $[p(x'|t=a)]$ by $[p(y'|v, t \neq a)]$.

It seems that there are at least two methods to solve this problem. One is to delete all the irrelevant rows after obtaining the direct product product of P and Q. The other is to avoid unnecessary multiplications which yield such irrelevant rows. We call the first "filtering method"[4] and the second "prepartitioned Kronecker product" or "modified Kronecker product," which will be explained in the following sections.

(3) This is also considered as a case of functional relation, $x=f(y)$, viz when $y=(t_i, v_j)$, $x=t_i$, x is a condensation of y, and therefore a many-to-one mapping, a function.

(4) As for this method, see M. Tsuji "Evaluation Methods for Networks of Information and Decision Systems," Wasede Business Review No. 4, 1978.

V. Computational Methods: Modified Kronecker Product

The idea of this method is to avoid unnecessary multiplications of probabilities, resulting from the Kronecker multiplication of one Markov matrix by the whole of the other matrix. For given two matrices this will be avoided by first partitioning them into sub-matrices according to indices of input messages and applying the Kronecker product only to the pairs of such matrices with the same index. The method will be explained for the cases introduced in the previous section.

When two processors P and Q receive an identical input message (as shown in Fig. 4. 1), P and Q are partitioned according to its common index,

$$(5.1)\qquad P=\begin{pmatrix} P^1 \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ P^i \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ P^m \end{pmatrix} \qquad Q=\begin{pmatrix} Q^1 \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ Q^i \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ Q^m \end{pmatrix}$$

where $P^i \equiv [p(x'|x=a_i)]$ and $Q^i \equiv [P(y'|x=a_i)]$.

Note that P^i and Q^i are row vectors which represent, respectively the conditional probabilities of x' and y' given $x = a_i$, The combined matrix resulted from P and Q, denoted by R, can be obtained by applying row-wise Kronecker products as follows:

$$(5.2)\qquad R=P\otimes Q=\begin{pmatrix} P^1\times Q^1 \\ \cdots\cdots \\ \\ \cdots\cdots \\ P^i\times Q^i \\ \cdots\cdots \\ \\ \cdots\cdots \\ P^m\times Q^m \end{pmatrix}$$

Symbol $\otimes$ means modified Kronecker product. This modification is necessary to avoid products such as $P^i \times Q^j$ for $i \neq j$.

The case of functional relation between messages can be treated similarly. A slight difference arises from the fact that processor Q receives two input messages t and v as illustrated in Fig. 4. 3. Matrix P is partitioned into m row vectors while matrix Q into m sub-matrices, i. e.,

$$(5.3)\qquad P=\begin{pmatrix} P^1 \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ P^i \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ P^m \end{pmatrix} \qquad Q=\begin{pmatrix} Q^1 \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ Q^i \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ Q^m \end{pmatrix}$$

where $P^i \equiv [P(x'|t=a_i)]$ and $Q^i \equiv \begin{pmatrix} P(y'|t=a_i, v=b_1) \\ \vdots \\ P(y'|t=a_i, v=b_n) \end{pmatrix}$

The combined matrix R can be computed as given by (5. 2. 2). It is important to note that the Kronecker product $P^i \times Q^i$ forms row vector by matrix multiplication.

We have a more general example of mixed coupling (partly overlapping) when P and Q share an identical input message while each receives an individual message, as illustrated by Fig. 5. 1. For instance, two members P and Q in a firm's department of finance specialize in stocks and bonds, respectively.

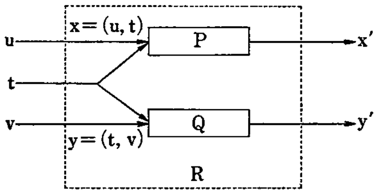

Fig. 5. 1 Example of Mixed Coupling

P receives his input from the analyst of stocks who sends him, as the message u, information about future prices and yields of individual stocks. Similarly, Q receives his input message v about future bond prices from the inquirer of bonds. But, in addition, both receive information about some general characteristics of future economy. The tasks of P and Q are to decide the dollar volume of funds to be invested in stocks and in bonds, respectively.

Since P and Q receive more than one input message, both are partitioned into m sub-matrices, i. e.,

$$(5.\ 4)\qquad P=\begin{pmatrix} P^1 \\ \cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots \\ P^i \\ \cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots \\ P^m \end{pmatrix} \qquad Q=\begin{pmatrix} Q^1 \\ \cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots \\ Q^i \\ \cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots \\ Q^m \end{pmatrix}$$

where $$P^i\equiv\begin{pmatrix} P(x'\,|\,t=a_i,\, u=b_1) \\ \vdots \\ P(x'\,|\,t=a_i,\, u=b_k) \end{pmatrix} \text{ and } Q^i\equiv\begin{pmatrix} P(y'\,|\,t=a_i,\, v=c_1) \\ \vdots \\ P(y'\,|\,t=a_i,\, v=c_n) \end{pmatrix}$$

The combined matrix R, which represents conditional probabilities of x' and y' given t, u and v, is created by augmenting matrix by matrix Kronecker product $P^i \times Q^i$ for $i=1, \cdots\cdots, m$, as given by (5. 2).

VI. Some More Complex Networks: Multiple Processors in Parallel

We have shown that the ordinary and modified Kronecker products are applied to deal with the cases of two processors coupled in parallel with one or two input messages. We will show in this section that the same method can be used for the case of more than two processors coupled in parallel.

There exist 16 kinds of network formulation when we have three processors with three input messages coupled in parallel, as summarized in Fig. 6. 1. The simplest and easiest formulation is the first one. Since three inputs are unrelated, the combined matrix will be obtained by using the ordinary Kronecker multiplication twice. The Kronecker product is associative as discussed in section II, and therefore the order of multiplication is not important, i. e.,

$$(6.\ 1)\qquad (P_1\times P_2)\times P_3=P_1\times (P_2\times P_3).$$

Case (2) and (3) in Fig. 6. 1 are simple extension of the cases with two processors coupled in parallel. After combining P_1 with P_2 by means of the modified Kronecker product, we apply the ordinary Kronecker multiplication to the combined matrix and P_3. The resultant matrix R may be expressed by

$$R=(P_1 \otimes P_2)\times P_3. \tag{6. 2}$$

Although case (16) looks very complicated, the solution is quite straightforward. All three processors receive the same input messages x_1, x_2, and x_3. Suppose that x_1, x_2, and x_3 range over the indices from 1 to n_1, 1 to n_2, and 1 to n_3, respectively and the rows of P_1, P_2, and P_3 are appropriately arranged in the same lexicographic order as (1, 1, 1) (1, 1, 2), ……, (1, 1, n_3) ……, (i, j, k) ……, (n_1, 1, 1) ……, (n_1, n_2, n_3). Then, row vector wise Kronecker products give us the final resultant matrix R,

$$R=\begin{pmatrix} P_1^{111}\times P_2^{111}\times P_3^{111} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ P_1^{11n_3}\times P_2^{11n_3}\times P_3^{11n_3} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ P_1^{ijk}\times P_2^{ijk}\times P_3^{ijk} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ P_1^{n_1n_2n_3}\times P_2^{n_1n_2n_3}\times P_3^{n_1n_2n_3} \end{pmatrix} \tag{6. 3}$$

where

$$P_1^{ijk}\equiv[P(x'_1|x_1=a_i,\ x_2=b_j,\ x_3=c_k)]$$
$$P_2^{ijk}\equiv[P(x'_2|x_1=a_i,\ x_2=b_j,\ x_3=c_k)]$$
$$P_3^{ijk}\equiv[P(x'_3|x_1=a_i,\ x_2=b_j,\ x_3=c_k)]$$

We will examine case (4) in detail because it requires a new operation in addition to the modified Kronecker multiplication. Whenever two processors receive two sets of input messages, one identical and the other different as P_2 and P_3, we have to rearrange the rows of such matrices in an appropriate lexicographic order for later multiplication of matrices. Suppose again that P_1, P_2, and P_3 are matrices of orders n_1 by n'_1, $n_1 n_2$ by n'_2 and $n_2 n_3$ by n'_3, respectively. First, we partition three matrices in the following way:

$$(6.\ 4)\quad P_1=\begin{pmatrix} P_1^1 \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ P_1^i \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ P_1^{n_1} \end{pmatrix} \quad P_2=\begin{pmatrix} P_2^1 \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ P_2^i \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ P_2^{n_1} \end{pmatrix} \quad P_3=\begin{pmatrix} P_3^1 \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ P_3^j \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ P_3^{n_2} \end{pmatrix}$$

where
$$P_1^i \equiv [P(x_1' \mid x_1=a_i)]$$
$$P_2^i \equiv \begin{pmatrix} P(x_2' \mid x_1=a_i,\ x_2=b_1) \\ \vdots \\ P(x_2' \mid x_1=a_i,\ x_2=b_{n_2}) \end{pmatrix}$$
$$P_3^j \equiv \begin{pmatrix} P(x_3' \mid x_2=b_j,\ x_3=c_1) \\ \vdots \\ P(x_3' \mid x_2=b_j,\ x_3=c_{n_3}) \end{pmatrix}$$

Secondly, we combine first two processors, P_1 and P_2 using the modified Kronecker product,

$$(6.\ 5)\quad P_1 \otimes P_2=\begin{pmatrix} P_1^1 \times P_2^1 \\ \cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots \\ P_1^i \times P_2^i \\ \cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots \\ P_1^{n_1} \times P_2^{n_1} \end{pmatrix}$$

where
$$P_1^i \times P_2^i=\begin{pmatrix} P(x_1',\ x_2' \mid x_1=a_i,\ x_2=b_1) \\ \vdots \\ P(x_1',\ x_2' \mid x_1=a_i,\ x_2=b_{n_2}) \end{pmatrix}$$

Note that $P_1^i \times P_2^i$ is ordered in such a way that x_1 is fixed for a_i and x_2 changed over from b_1 to b_{n_2}. Since P_3^j is ordered in an incompatible way with this, $P_1 \otimes P_2$ has to be rearranged as follows:

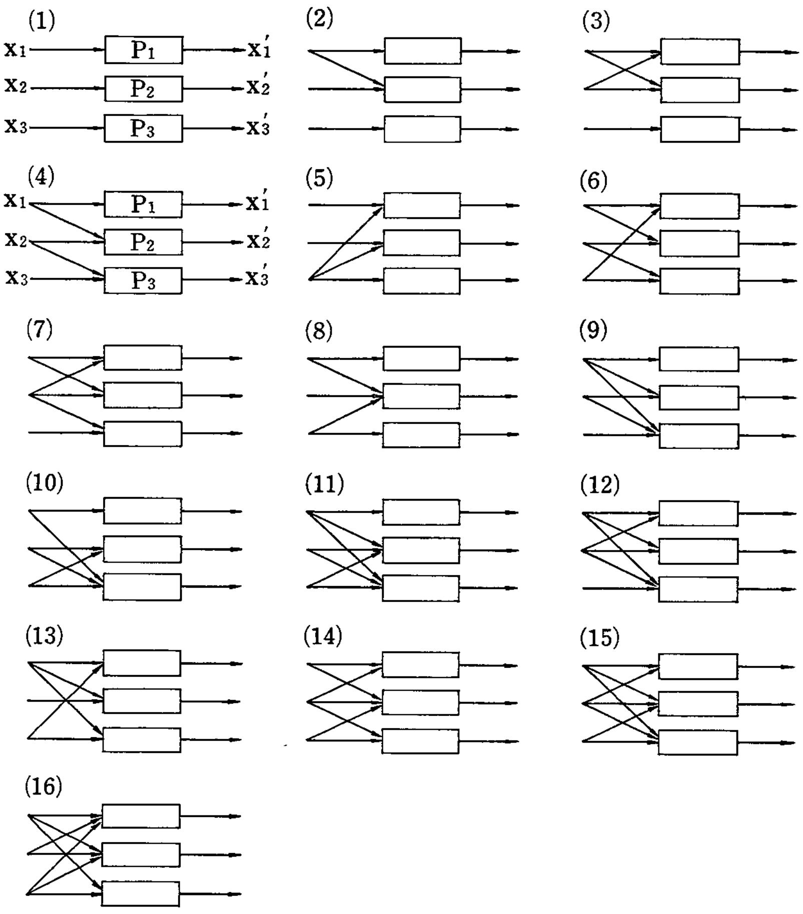

Fig. 6. 1 Some More Complex Networks: Three Processors

(6. 6) $$(P_1 \otimes P_2) = \begin{pmatrix} (P_1 \times P_2)^1 \\ \vdots \\ (P_1 \times P_2)^j \\ \vdots \\ (P_1 \times P_2)^{n_2} \end{pmatrix}$$

where $$(P_1 \times P_2)^j \equiv \begin{pmatrix} P(x_1', x_2' | x_1 = a_1, x_2 = b_j) \\ \vdots \\ P(x_1', x_2' | x_1 = a_{n_1}, x_2 = b_j) \end{pmatrix}$$

The final operation is taken to combine the above matrix with P_3, which gives us the final result R,

$$(6.7)\quad R=(P_1\otimes P_2)\otimes P_3=\begin{pmatrix} (P_1\times P_2)^1\times P_3^1 \\ \cdots\cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots\cdots \\ (P_1\times P_2)^j\times P_3^j \\ \cdots\cdots\cdots\cdots \\ \vdots \\ \cdots\cdots\cdots\cdots \\ (P_1\times P_2)^{n_2}\times P_3^{n_2} \end{pmatrix}$$

where
$$(P_1\times P_2)^j\times P_3^j\equiv\begin{pmatrix} P(x_1', x_2', x_3'\,|\,x_1=a_1,\ x_2=b_j,\ x_3=c_1) \\ \vdots \\ P(x_1', x_2', x_3'\,|\,x_1=a_1,\ x_2=b_j,\ x_3=c_{n_3}) \\ \vdots \\ P(x_1', x_2', x_3'\,|\,x_1=a_i,\ x_2=b_j,\ x_3=c_k) \\ \vdots \\ P(x_1', x_2', x_3'\,|\,x_1=a_{n_1}, x_2=b_j,\ x_3=c_1) \\ \vdots \\ P(x_1', x_2', x_3'\,|\,x_1=a_{n_1}, x_2=b_j,\ x_3=c_{n_3}) \end{pmatrix}$$

In any complex case where more than three processors are coupled in parallel, we can apply the same procedure repeatedly until we incorporate all the processors under consideration. First, we examine two processors and check if their inputs are related or not. If related, we use the modified Kronecker multiplication, and if not, the ordinary Kronecker multiplication. The Kronecker multiplication combines two processors and produces a single resultant matrix. Next, we choose third processor and examine to see if the inputs to this and the combined processors are related. If necessary, the rows of matrices are arranged in an appropriate lexicographic order for the purpose of later Kronecker multiplication. From these observations, we have the following theorem:

Theorem 6.1: Multiple processors coupled in parallel can be combined by the ordinary and modified Kronecker products.

VII. Series-Parallel Coupling and More Complicated Networks

Firstly, suppose that two processors are coupled as shown in Figure 7. 1. Processor Q receives two kinds of input messages: one from the outside and the other from processor P. How do we compute the combined transition matrix which represents the conditional probabilities of given x_1 and x_2 in this network?

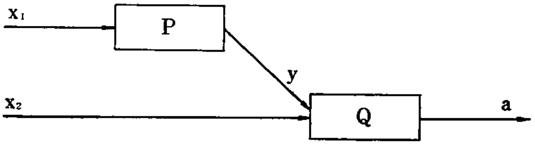

Figure. 7. 1 Two Processors: Two Inputs

We assume, for a while, that x_1, x_2, and y are not related. Although processors P and Q are coupled in series, we cannot use the conventional multiplication of product P • Q because of the existence of input message x_2. To deal with this situation we may consider that Q receives x_2 through a (fictitious) processor rather than from the outside. This fictitious processor can be represented by a deterministic and hence identity matrix with the dimension of x_2. Thus, the above network may be rewritten as shown in Figure 7. 2. Since processors P and Ix_2 are coupled in parallel and their inputs are not related, the resultant matrix of P and Ix_2 are computed by using the direct product,

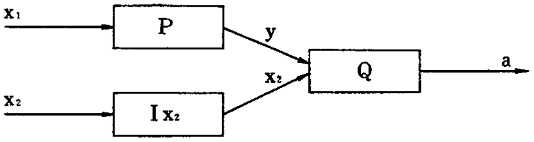

Figure. 7. 2 Two Processors: A Fictitious Processor

$$P \times I_{x_2} = \begin{pmatrix} P & & & \\ & P & & \\ & & \ddots & \\ & & & P \end{pmatrix} \tag{7. 1}$$

Obviously, the combined processor of P and I_{x_2} is connected with processor Q in series. The ordinary matrix multiplication will give us the overall transition matrix,

(7. 2) $\quad R=(P\times I_{x_2})Q$

In the case of related inputs, the modified Kronecker product will be used to obtain the resultant matrix. A fictitious processor must be inserted in an appropriate position as in the unrelated case. The overall matrix will be provided as follows:

(7. 3) $\quad R=(P\otimes I_{x_2})\otimes Q$

Secondly, we consider a network which is little more complex

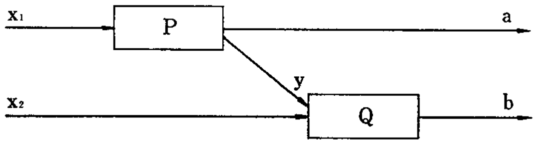

Figure. 7. 3 Two Processors: Two Outputs

than the previous one, as shown in Figure 7. 3. The only difference is that processor P produces two kinds of input messages in this case. Another fictitious processor, denoted by Ia, will be inserted next to processor P, as illustrated in Figure 7. 4.

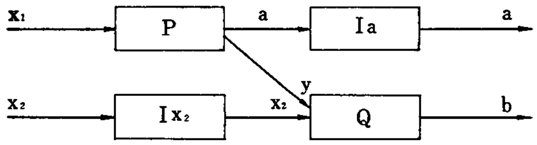

Figure. 7. 4 Two Processors: Two Fictitious Processors

If the input messages are independent with one another, the following computation will be made to get the overall matrix,

(7. 4) $\quad R=(P\times I_{x_2})(I_a\times Q).$

On the other hand, if the input messages are related, the modified Kronecker product will be applied as follows:

(7. 5) $\quad R=(P\otimes I_{x_2})(I_a\otimes Q).$

As we have seen in the above examples, the necessity of adding a fictitious processor arises whenever a processor receives at least two kinds of input messages (one from the outside and the other from another processor) or a processor sends two kinds of output messages (one to the outside and the other to another processor).

To find relationships among processors in a network, the following definition may be useful.

Definition 7. 1. For a given network of processors, binary matrix B, called network matrix, is definied such that its elements

$$(7.\ 6)\quad X_{ij}=\begin{cases}1 & \text{if processor } P_i \text{ sends messages to } P_{,j}\\ 0 & \text{otherwise}\end{cases}$$

for $i=0, 1, 2, \cdots\cdots, m$, and $j=1, 2, \cdots\cdots, m+1$, where m is the number of processors in the network. P_0 represents the part of environment or the state of nature on which organization members make observations. P_{m+1} stands for the part of environment to which final actions of an organization are directed.

Thirdly, consider the network of processors which is represented in Figure 7. 5. The network matrix for this complex network will be obtained readily by following Definition 7. 1,

$$(7.\ 7)\quad B=\begin{pmatrix}1&1&1&0&0&0&0&0&0&0\\0&0&1&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0\\0&0&0&0&1&1&0&0&0&0\\0&0&0&0&0&1&1&0&0&0\\0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1&1&1\\0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&0&1\\0&0&0&0&0&0&0&0&0&1\end{pmatrix}$$

Note that l's in rows of network matrix B imply the communication flows originated from those processors and l's in colums mean the flows directing to those processors.

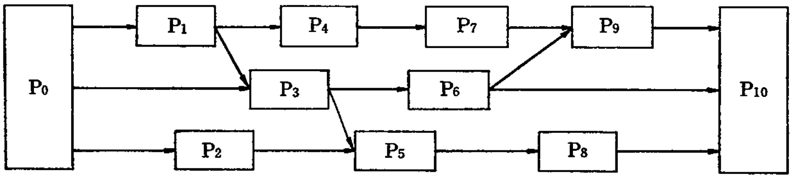

Figure. 7. 5 Series-Parallel Coupling

Where should we add fictitious processors to get the overall transition matrix by using the method developed in the previous sections? It seems that there are two ways to proceed:

(1) To choose at random any two processors in a network and to augument the number of processors one at a time until we are left with one processor.

(2) To start with the processors which receive messages from the environment and to form the resultant matrix step by step by following the arrowed direction.

Since both methods are essentially the same, we will develop only the latter in this paper.

For the simplicity sake, we assume that input messages are not related unless they are provided by the same processor. First, we pick up those processors which receive input messages only from P_0, i. e., P_1 and P_2 in this example. These processors are called processors at the first stage. Because P_3 receives input messages from P_0 as well as P_1, a fictitious processor is necessary between P_0 and P_3, and is denoted by I_{03} in Figure 7. 6. Then, the processors at the first stage consist of P_1, I_{03} and P_2 in this example. Using the method developed in the previous section, the combined matrix will be obtained as

(7. 8) $$R_1 = P_1 \times I_{03} \times P_2$$

Secondly, after the calculation at the first stage, we accumulate the environment by adding the processors at the first stage to the original environment. Then, as we have done at the previous stage, select those processors which receive input messages from the new environment. They are P_3 and P_4 which form the processors at the second stage. They do not include P_5 because P_5 receives input messages from P_2 as well as P_3 which itself is a member at the second stage. Therefore, a fictitious processor, denoted by I_{25}, is added between P_2 and P_5. Thus, we get the combined matrix at the second stage,

(7. 9) $$R_2=(P_4 \otimes P_3)\times I_{25}$$

Since R_1 and R_2 are coupled in series, they are combined by using the ordinary multiplication of product,

(7. 10) $$R_{12}=R_1 \cdot R_2$$

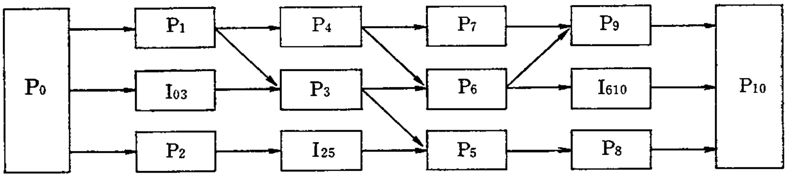

Figure. 7. 6 Series-Parallel Coupling : Four Stages

Similarly, by applying the multiplication of modified Kronecker product, we have the combined matrices at the third and fourth stages,

(7. 11) $$R_3=P_7 \otimes P_6 \otimes P_5,$$

$$R_4=P_9 \otimes I_{610} \otimes P_8$$

Then, the overall transition matrix is readily given by

(7. 12) $$R_{1234}=R_1\ R_2\ R_3\ R_4$$

Finally, we summarize the above procedure as "algorithm," by referring to binary number of a network matrix.

Step 1. Specify the current environment (Start with P_0 at the first stage).

Step 2. Select those processors which have 1's in row P_0.

Step 3. Look at the columns of selected processors and check if there is no 1 in other rows. Those processors which have 1's in other rows receive input messages from other than the current environment. If not, go to Sept 5. If any, go to Sept 4.

Step 4. Add a fictitious processor between the current environment and those processors which receive input messages from other than the environment.

Step 5. Combine those processors which have 1's in row of current environment and, if any, fictitious processors, by using the method developed in this paper.

Step 6. Accumulate the current environment by adding the combined processors to the environment.

Step 7. Go back to step 1 if any processor is left uncombined and stop otherwise.

VIII. Concluding Remarks

An organization consisting of decision makers, staffs, and data processing and communication equipments is viewed as a network of processors or functions. The gross benefit of a purposive organization is a function of "events" (inputs from outside) and "actions" (outputs to the outside). All other in-and outputs of the processors are considered internal messages.

An optimal network yields maxinal expected difference between benefit and managerial costs of task performance, given the benefit and cost functions and the probabilities of events. This expected difference is determined by the network's overall transition matrix stating the conditional probability of each (joint) action given each event. This paper has shown that various types of connections between processors require different kinds of computations:

(1) Coupling in series: use conventional multiplication of a sequence of matrices;

(2) Coupling in parallel when all pairs of possible input messages can occur: use Kronecker product of matrices;

(3) Coupling in parallel when not all pairs of possible input mes-

sages can occur: use modified Kronecker product.

The tools developed in this paper will be tried out as to the computational efficiency. They will also help to compare the statistical "informativeness" for some pair of networks regardless of their user.

References

1. Bellman, R, *Introduction to Matrix Analysis,* Second Edition (Prentice-Hall, 1970).
2. Booth, T. L., *Sequential Machines and Automata Theory* (John Wiley and Sons, 1968).
3. Drenick, R. F. and A. H. Levis, *A Mathematical Theory of Organization Part I: Centralized Organization* (Polytechnic Institute of New York, 1974).
4. Marschak, J., "Efficient Organizational Design," in *Economic* Theory for Economic Efficiency: Essays in Honor of Abba P. Lerner (in press).
5. Marschak, J., and R. Radner, *Economic Theory of Teams* (Yale University Press, 1972).
6. Pease, M. C. III, *Methods of Matrix Algebra* (Academic Press, 1965).
7. Tsuji, M., "Evaluation Methods for Networks of Information and Decision Systems," *Waseda Business Review* No. 4, 1978.