AN ABSTRACT OF THE THESIS OF

JOZEF M. DUMON for the degree DOCTOR OF PHILOSOPHY
(Name of student) (Degree)

in Industrial Engineering presented on August 8, 1974
(Major Department) (Date)

Title: AN RPM NETWORK APPROACH TO THE INTEGRATION OF THE ACCOUNTING INFORMATION SYSTEM AND THE LINEAR PROGRAMMING DECISION MODEL

Abstract approved: Redacted for Privacy
Dr. Stephen F. Love

The use of the Resource Planning and Management (RPM) network technique as a common tool of communication between the information user (e. g. manager, accountant) and the model builder or management scientist as well as between the systems analyst-programmer and the management scientist is examined.

The premise is that the discrepancy between an information system and a decision model need not exist. By explicitly joining, in network form, the cause and effect relationship recognized in double entry accounting with that in linear programming, the accounting system is informationally integrated with the linear programming decision model.

The role of the systems analyst-programmer in effecting a realization of the model on electronic data processing equipment is

recognized. The RPM network technique is used as a tool to derive a data structure useful to the systems analyst-programmer in the making of this integration.

The development of a programming system and its application to a hypothetical illustration demonstrates the feasibility of the approach.

An RPM Network Approach to the Integration of the Accounting Information System and the Linear Programming Decision Model

by

Jozef M. Dumon

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

June 1975

APPROVED:

Redacted for Privacy

Professor of Industrial Engineering
in charge of major

Redacted for Privacy

Head of Department of Industrial Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented August 8, 1974.

Typed by Lyndalu Sikes for Jozef M. Dumon

ACKNOWLEDGMENTS

I should like to express my sincere thanks to Dr. Stephen Love, for his guidance and help generously provided during the course of this dissertation. Special thanks are due to Dr. Michael Inoue for his encouragement and suggestions, that resulted in many improvements. I wish to extend my special thanks further to all members of my graduate committee, for their helpful suggestions and careful reading of this dissertation.

TABLE OF CONTENTS

Chapter	Page

LIST OF FIGURES

Figure		Page

LIST OF APPENDIX FIGURES

LIST OF TABLES

LIST OF APPENDIX TABLES

AN RPM NETWORK APPROACH TO THE INTEGRATION OF THE ACCOUNTING INFORMATION SYSTEM AND THE LINEAR PROGRAMMING DECISION MODEL

I. INTRODUCTION

With the development of larger computers, increased computational speed and storage media, the promise for the development and use of quantitative management models became brighter. However, recent surveys indicate that OR/MS teams have fallen short in the fulfillment of these promises. Urban (1974) reports that . . .

> In the period January 1971 to June 1973 Management Science: Applications contained over 150 articles, but less than 3 percent represented implementation in an organization (used more than once) and only 15 percent more were applied even once in a real decision situation.

Unavailability or inaccessibility of data, needed in the several phases of a model development, has been an important concern of the people involved. We can hear it in the voice of the manager (Grayson, 1973):

> A manager will ordinarily use data or a management science tool only if both are conveniently, speedily accessible. If he is told that the needed data are buried in another part of the organization, or that they must be compiled, or that the model must be created, nine times out of ten he will say "Skip it". I did, ten times out of ten.

From the management scientist side, we hear (Vazsonyi, 1973):

> Getting data for OR/MS [Operations Research/Management Science] work is one of the big headaches. Can MIS

> [Management Information System] produce the data? . . .
> In theory all MIS work must start with building a model. In fact, we are often told that it is impossible to build a MIS without a model. In practice, it is hard to follow this precept, and in reality the model is usually ignored. Why is this? . . .

And this is what the accountant says (Greynolds, 1972):

> The determination of cost input information for various operations research models is often as important as the development of the model itself. Many of the people who generate the input data have limited knowledge of the requirements for the model, while the operations research specialist may not have full understanding of the origin of the information used in the model.

Embarking on an OR project impounds two interrelated sub-projects (Hartley, 1968): the model construction and initial implementation phase, and the recurrent solution and model updating phase. Figure 1.1 shows activities (corresponding to squares), and related information (corresponding to circles) produced and used for the model construction phase. Figure 1.2 represents the solution and model updating pattern.

Preferably, the cause and effect analysis is performed using a team approach, where parties with different disciplines and interest indicate the relevant factors for the given problem situation (Inoue and Riggs, 1971). The factor classification process involves the sorting of the above factors into controllable and uncontrollable ones; the controllable into significant and non-significant ones

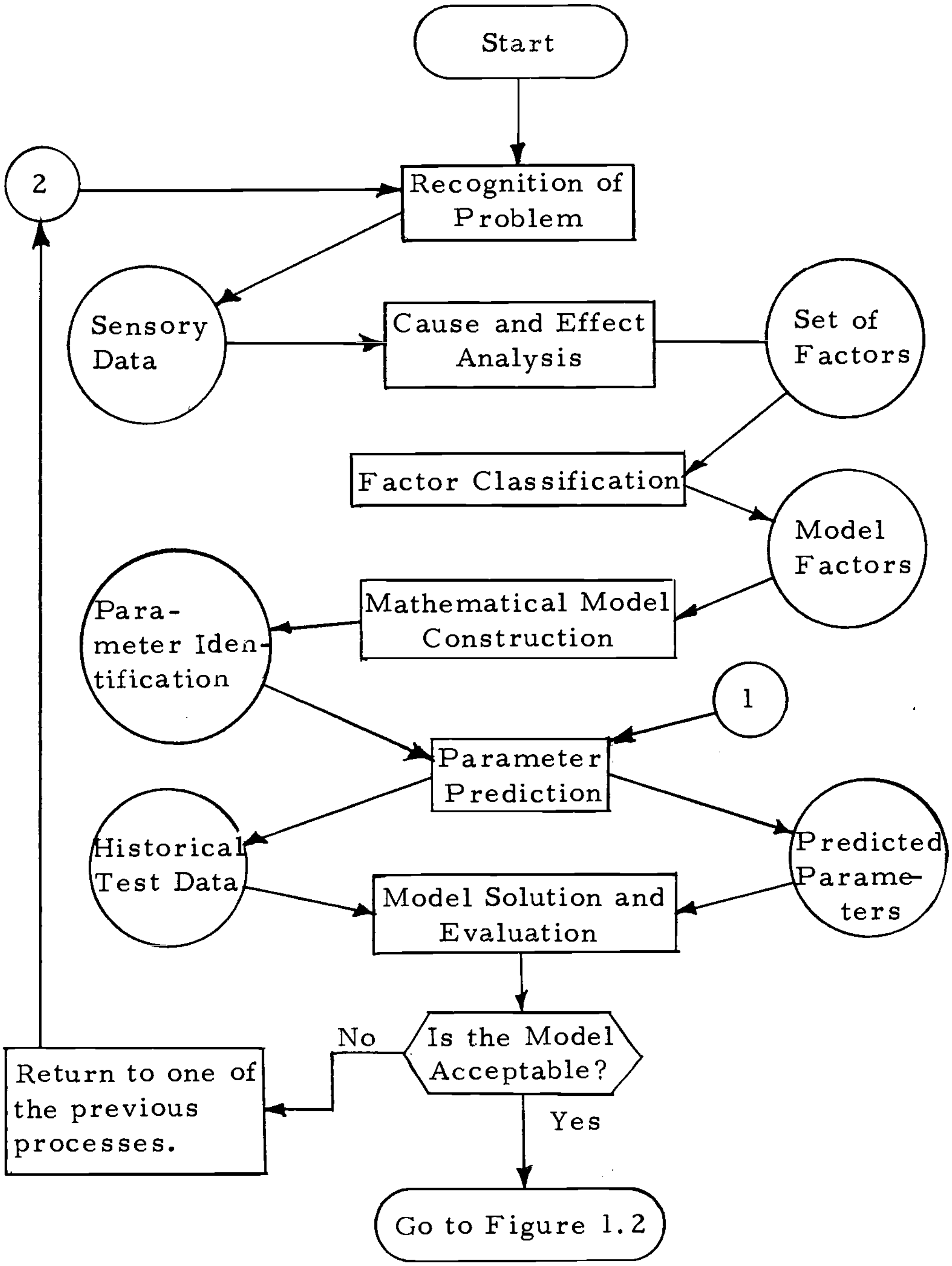


Figure 1. 1. Model construction and initial implementation phase; Related information inputs and outputs.

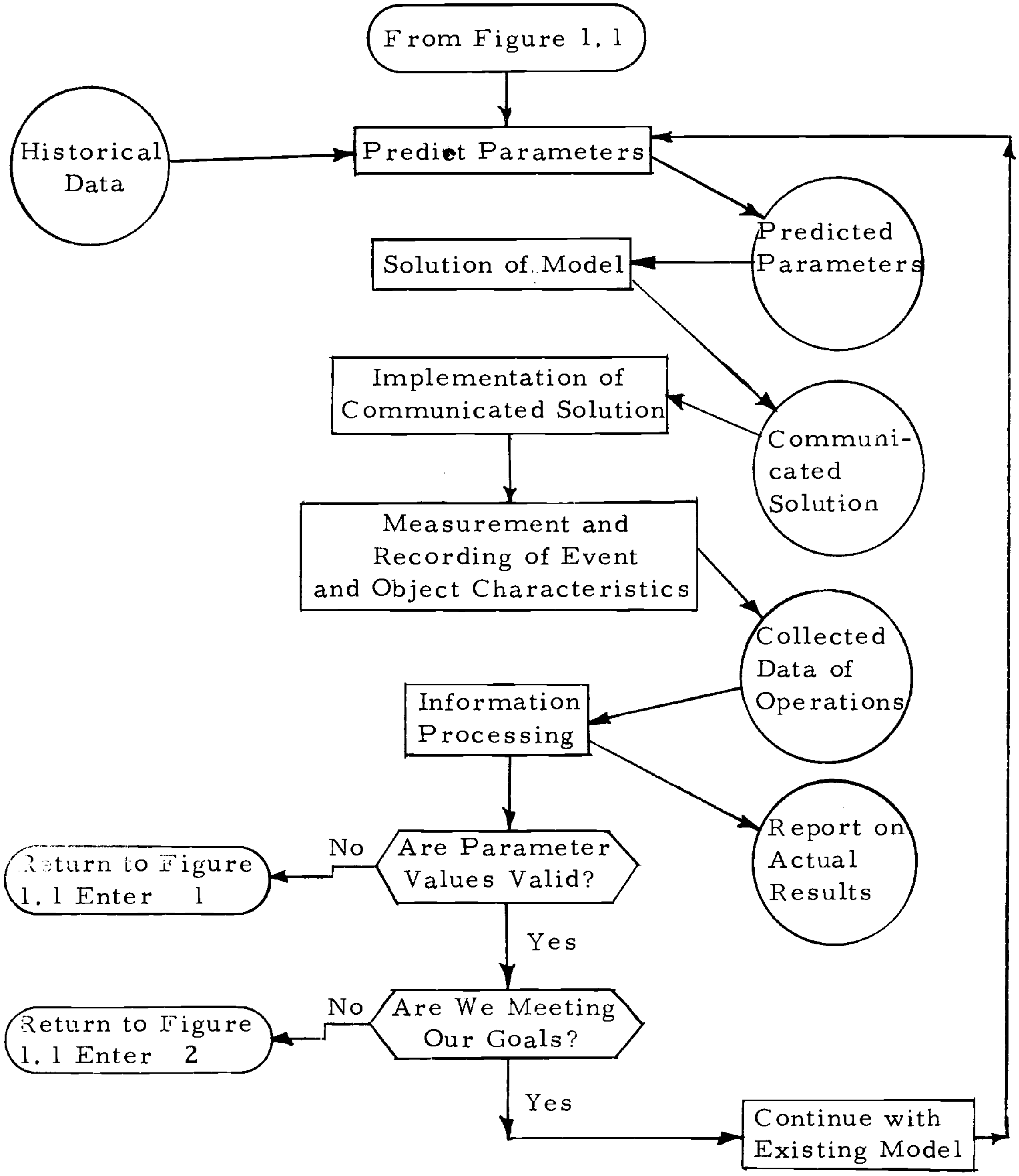


Figure 1.2. Recurrent solution and updating phase; Related information inputs and outputs.

(e. g., by the use of Pareto analysis); and the sorting of significant factors into quantifiable and non-quantifiable ones.

The model then indicates the parameters that need to be estimated. These values and the model in general are then tested for validity by the use of test data.

The "recurrent solution control process" (Figure 1.2) refers to the processes necessary to compute solutions and to evaluate the validity of the assumptions and the relationships over successive time periods. It is clear that this process is only relevant for recurring decisions, with a more or less stable pattern, such as inventory models, the product mix problems etc.... For one time decisions, such as the optimal location of a plant, one doesn't need to update the model again, once the decisions are made.

We must recognize that the problems of information specialists are also the problems of management scientist, and vice versa. In his fact-finding analysis and design functions, the information and data systems analysts reduce the complexity of a real-world system, so that selected factors become explicit in the information system. On the other hand, the management scientist wants to help managers make decisions more explicitly and better by using scientific approaches. Therefore, management science needs are to be incorporated in the design of an information system or else the management scientist will go on calling for inaccessible or non-existant or uncompiled data for his operations research models, and hence managers will continue saying "Skip it".

This deficiency has been prevalent for a long time. Pioneering work done by Ijiri, Levy and Lyon (1963), Charnes, Cooper and Ijiri (1963), Ijiri (1965, 1967) incorporated accounting variables into LP decision models. Since then, some information related theories have also developed that formalize different aspects of the operating subsystem and the information subsystem in a decision making context. (See Appendix I for a short description of theories and concepts used.)

Yet, little effort has been directed to the effective use of networks in coping with these communication problems.

The importance of the use of networks was early stated by Charnes, Cooper and Ijiri (1963), when they wrote:

> Further research, especially on the network aspects of these accounting models, will probably be necessary in order to devise suitable artifacts that will avoid unwieldiness and achieve the efficiency that will undoubtedly be needed when actual applications to large-scale management problems are essayed.

Charnes and Cooper (1967) made a series of sketches and suggestions for improved approaches to problems in management planning and control, but limited their presentation to a relationship between PERT/ CPS networks and the booking of corresponding account transactions.

The main purpose of this dissertation is to develop a basis by which the Resource Planning and Management (RPM) network technique[1] can be used as a communication device between:

1. the manager and the management scientist; and between
2. the systems analyst-programmer and the management scientist.

A tree structured network, widely used in computer science to represent the hierarchy of data in systems, will be used here as a means to represent the hierarchy of accounts[2]. Such a structure will be presented in a RPM network form and used to study the relationship between the accounting information system and the RPM-LP decision network. A second use of the RPM network technique in the communication process between manager and modeler can be made by developing a methodology to derive an LP model from a network representing the monetary flows between accounts. Finally, a methodology will be developed to derive a data structure for RPM networks in general, and for the tree structured network as a special simplified case.

---

[1]The following references describe the RPM concepts and principles: (Inoue and Riggs, 1972), (Chen, 1973), (Inoue and Riggs, 1975).

[2]A similar hierarchical structure for accounts of an accounting system was presented at the 44th National Conference of ORSA by Keast (1973), who reports on its actual use for updating the accounting system for forecasted transactions in the Westinghouse Division of Aerospace and Electronic Systems.

In the next chapter, we will analyse the existence of the cause and effect relationship in an accounting system inherent in the (linear) aggregating subsystem of a total information system. This cause and effect relationship in accounting described by Ijiri (1965), will be related to the cause and effect relationship in linear programming models for decision making, by the use of the RPM network technique, developed by Inoue and Riggs (1972, 1975), that recognizes this relationship explicitly. By developing a tree type RPM network to represent the conventional set of accounts and closing transactions among them, it becomes possible to unify the cause and effect relationship of the LP model with that of the accounting system. Using the LP model as a point of departure, we then investigate the level of influence that the LP model has in changing the status of the accounting information system.

In Chapter III, we will use the accounting information system as a point of departure for constructing an LP decision model. Given the beginning balances and the transaction amounts for a given operating period, an interaccount flow network will be constructed according to the RPM conventions. This chapter will be primarily devoted to converting this IAF network to a standard RPM-LP model. The network obtained through this derivation process will have the characteristics of a Corporate Model, because of its close relationship with the accounting variables.

Chapter IV studies how the results of the LP model can be converted into accounting transactions. The purpose is to update the beginning balances of an accounting information system, so that effects of the results from the LP model can be studied together with the predicted results of other decisions, external to the decision field of the LP model. These effects will be reported through the financial statements.

In Chapter V, we shall study how the RPM network technique can be used to derive a data structure for the (computer) programming of management science algorithms (here applied to the primal-dual simplex algorithm (MINIT) developed by Llewelyn (1964)), and for its related data requirements (here the accounting information system). A generalized programming system was developed that accepts any LP model, accounting system in hierarchical format, and conversion matrix information. The output of the system is reported through the three conventional accounting statements: the balance sheet, the income statement and the funds flow statement. Technical information about the use of the system is described in Appendix III.

Finally, Chapter VI discusses the limitations and suggested future expansions of this study.

II. CONSISTENCY BETWEEN THE COST INFORMATION SYSTEM AND THE LP MODEL

As discussed in Appendix I, the data collection and reporting can be considered as the two ends of the information process. The collection of data involves a primary measurement process and a data gathering process. The primary measurement process, consisting of assigning values to selected characteristics of events, objects or activities, will not be discussed in this dissertation. However, the derived measurement process that applies data transformation functions on primary measurements will concern us insofar as it pertains to the conventional accounting computations.

Usually, the primary measurement process leads to a data gathering on source documents; the aggregated versions of these documents form the subledgers of an accounting system; the accounting system itself represents then the monetary values of the equities and assets of an entity and monitors changes in these values. In this dissertation, we shall assume that the aggregation process has reached the level of an accounting system.

In the first section of this chapter, we shall look at accounting transactions as being based on cause and effect relationships. In the second section, we shall study the fundamental equations for several accounting reports. By disaggregating the balance sheet, it becomes possible to build a hierarchical accounting (HA) network. The income statement equation is represented as a subset of this HA network. In the third section, we shall demonstrate with a simple example the use of the HA network to determine the degree to which a LP model embraces the total set of company activities.

Cause and Effect Relationship in Accounting

The fundamental accounting equation indicates that A(ssets) = L(iabilities) + O(wners' equities). The left side of the equation can be further partitioned by the type of asset (or resource) and the right side by the type of claimant (or source), as is partly done on the balance sheet.

The strategic, operational and control actions of managers cause flows in the assets and equities. The concept of flow is used to stress the cause and effect relationship that typifies an accounting transaction. An increment in one account has to be caused by an equal decrement in another account. In order to clarify the causal aspect of double entry, we shall deal only with "simple transactions". Compound transactions can be defined as having more than one debit (increment) or credit (decrement) entry. However, compound transactions can be decomposed into a set of simple transactions, so that the causality discussion can be generalized, without loss of reality.

The accounting theory has witnessed two approaches to its double-entry mathematics (Ijiri, 1967). Let us consider "A" to be the set of all asset accounts $\{a_i\}$, and Y_A to be the sum of the asset account balances $\{Y_{a_i}\}$. Similarly, let E be the set of all equity accounts $\{e_i\}$, and Y_E be the sum of the equity account balances $\{Y_{e_i}\}$. Then the asset partitioning and the equity partitioning are considered two classifications of a same total value:

1. a classification of the physical characteristics of the resources (assets).
2. a classification according to the claimants of the assets (equities).

By definition, we have

$$\sum_{i=1}^{m} Y_{a_i} = Y_A = Y_E = \sum_{j=1}^{n} Y_{e_j} \tag{2.1}$$

In the second approach (causal double-entry) only one classification is used. The assets and equities are brought into one classification $B = \{A, E\} = \{a_1 \ldots a_m, e_1 \ldots e_n\}$. The "sales of a finished product paid in cash" expresses the cause and effect relationship between a decrement in finished product inventory followed by an increment in cash for the same value. However, the cause and effect relationship may be obscured if we involve equity accounts in transactions. It is therefore useful to consider equity accounts as future cash decrements. In order to include this extension of assets, the syntax of accounting language has introduced the concepts of negative assets to refer to equity accounts (Ijiri, 1967). For example, if an entity purchases materials on credit, then we recognize the causal relationships between a future decrement in cash, expressed in the accounts payable, and a present increment in materials account. The other equity accounts can be interpreted in a similar way. First, the liability accounts can be considered as future decrements in cash, whose corresponding increments in present (positive) assets have occurred; second, the owner's equities accounts may also be considered as future decrements in cash, i.e., when the dividends are paid or in case an entity is considered for liquidation.

Algebraic Double Entry and the Hierarchic Accounting Network

Because of the need to recognize the causality of a transaction, the argument for classification should not be abandoned. After all, the causality approach to the double-entry bookkeeping system is based on "one" classification of positive and negative assets.

As an entity consists of positive and negative assets, the following algebraic rules should be applied to decide on a debit or a credit entry:

1. Use a debit entry for an increase in a positive asset (e.g., cash) and for a decrease in a negative asset (e.g., accounts payable). Note also that an expense should be debited because the expenses represent a (temporary) decrease of a stockholders' equity.
2. Use a credit entry for a decrease in a positive asset (e.g., cash) and for an increase in a negative asset (e.g., accounts payable). A converse argument, as made for the expenses, holds with respect to the revenue accounts.

Let the equation $Y_{A_1} = Y_{L_1} + Y_{O_1}$ correspond to the balance sheet at the end of period one or the beginning of period two, and $Y_{A_2} = Y_{L_2} + Y_{O_2}$ to the end-of-period two balance sheet, then

$$Y_{A_2} = Y_{A_1} + \Delta Y_A \qquad (2.2)$$

$$Y_{L_2} = Y_{L_1} + \Delta Y_L \qquad (2.3)$$

$$Y_{O_2} = Y_{O_1} + \Delta Y_O \qquad (2.4)$$

The change in assets (ΔY_A), caused by all of the transactions can result in a total increase (ΔY_A^+) or a total decrease (ΔY_A^-) in the total asset value. A similar notation holds for the liabilities (ΔY_L^+, ΔY_L^-) and for the owner's equity (ΔY_O^+, ΔY_O^-).

For the purpose of deriving the financial equation corresponding to financial statements other than the balance sheet, four types of transactions will be introduced expressing the causality of change between the balance sheet of period one and period two (Nicol, 1968; Murphy, 1970):

1. The "Revenue Transaction" (X_R) represents all transactions related to all kind of sales.
2. The "Expense Transaction" (X_E) represents all transactions related to production, marketing, research and development . . . .
3. The "Dividend Transaction" (X_D) represents the dividend drawing for stockholders.
4. The "Investment Transaction" (X_I) represents a change in owner's equities.

Each of these four transactions may give rise to a change in assets (ΔY_A) and/or a change in liabilities (ΔY_L).

Assuming that the entity engages in normal production and trading operations during the period, and investments and dividend drawings occurred, then

$$\Delta Y_O = X_R - X_E + X_I - X_D \qquad (2.5)$$

Because $Y_{A_2} = Y_{L_2} + Y_{O_2}$, and using (2.4) and (2.5), we can form the "Preclosing Trial Balance" at the end of period two, represented by

$$Y_{A_2} + X_E + X_D = Y_{L_2} + (Y_{O_1} + X_I) + X_R \tag{2.6}$$

The next step in the accounting process, is the closing and the derivation of the "Income Statement". This is done by subtracting X_E from both sides of equation (2.6):

$$Y_{A_2} + X_D = Y_{L_2} + (Y_{O_1} + X_I) + X_R - X_E \tag{2.7}$$

The description of $(X_R - X_E)$ represents the Income Statement.

Now, if we further subtract X_D from both sides of (2.7), then we have

$$Y_{A_2} = Y_{L_2} + (Y_{O_1} + X_I) + (X_R - X_E) - X_D \tag{2.8}$$

where $(Y_{O_1} + X_I) + (X_R - X_E) - X_D$ represents the Owner's Equity Statement.

Although there is some disagreement on how the Funds Flow Statement is to be defined, the most widely accepted defition is that of describing the changes in the net working capital (Horngren, 1965). The net working capital is defined as the excess of current assets over current liabilities. If we apply the balance sheet classification to equation (2.1), we have the identity

$$\sum_{i=1}^{m} Y_{a_i} = Y_A = Y_L + Y_O = \sum_{j=1}^{n} Y_{e_j}$$

Now, if we let

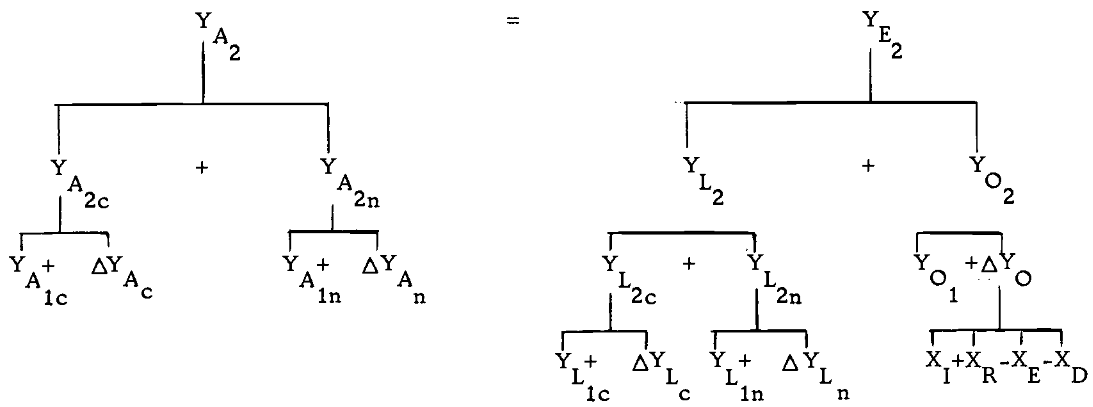
Y_{A_c} be the balance of current assets

Y_{A_n} be the balance of non-current assets

Y_{L_c} be the balance of current liabilities

Y_{L_n} be the balance of non-current laibilities,

the further subdivision of the balance sheet equation can be represented in the following manner:



The change in net working capital can now be expressed as $\Delta Y_{A_c} - \Delta Y_{L_c}$. If we substitute the handle of these two trees into their respective roots, we obtain the more expanded balance sheet equation:

$$(Y_{A_{1c}} + \Delta Y_{A_c}) + (Y_{A_{1n}} + \Delta Y_{A_n}) = (Y_{L_{1c}} + \Delta Y_{L_c}) + (Y_{L_{1n}} + \Delta Y_{L_n}) + (Y_{O_1} + X_I) + (X_R - X_E) - X_D \qquad (2.9)$$

If we subtract the end-of-period-one balance sheet equation, namely

$$Y_{A_{1c}} + Y_{A_{1n}} = Y_{L_{1c}} + Y_{L_{1n}} + Y_{O_1} \tag{2.10}$$

from equation (2.9), then we obtain:

$$\Delta Y_{A_c} + \Delta Y_{A_n} = \Delta Y_{L_c} + \Delta Y_{L_n} + X_I + (X_R - X_E) - X_D \tag{2.11}$$

By rearranging the terms of (2.11), the change in net working capital can be expressed as:

$$\Delta Y_{A_c} - \Delta Y_{L_c} = \Delta Y_{L_n} - \Delta Y_{A_n} + (X_R - X_E) + X_I - X_D \tag{2.12}$$

This change can then be studied as a function of

a. a change in non-current assets $(\Delta Y_{L_n} - \Delta Y_{A_n})$

b. income $(X_R - X_E)$

c. additions to or distribution from the owner's equity $(X_I - X_D)$.

This algebraic aggregate representation of several aspects of financial analysis draws our attention to the possibility of representing their detailed structure through a tree network. To accomplish such a tree structure in a Resource Planning and Management (RPM) context, a number of conventions will have to be introduced:

1. Each "resource" node is to be considered as a balance of an account. This means that in a detailed network, not

only should all ledger accounts be represented by the balance value, but all balance sheet entries are to be considered as aggregate accounts of lower level account balances. These resource nodes will also be referred to as "account nodes".

2. Each "activity" node represent a closing out transaction. This means that the positive (or negative) balance value of a lower level account, if one exists, is transferred to the nixt higher level asset (or equity) account. The transferred values are usually used as entries on the balance sheet once the level of balance sheet accounts is reached. These activity nodes will also be termed "closing transaction nodes".

3. The direction of the arrows connecting closing transaction and account nodes is upward on the asset side, and downward on the equity side of the account balance tree. The latter direction can be made upward by making the transferred values negative or by using a -1 coefficient on the arrows. The general convention is that an arrow leaving a resource node corresponds to a credit booking on an account, while an entering arrow refers to a debit booking.

4. The account nodes with corresponding accounts in the management information system are indicated by double circles on RPM networks.

Figure 2.1 shows an example of such a tree network for the conventional accounting system; this network will be called the hierarchical accounting tree (HA tree or RPM-HA tree).

Interfacing the RPM-HA Tree and the RPM-LP Network

In the first section of this chapter we indicated why causality can be called the backbone of the double-entry bookkeeping system. Because RPM-LP (Linear Programming) networks are a direct expression of the cause-and-effect relationship we should be able to relate them to the RPM-HA network. In this section we shall illustrate the basic integration approach on a simple example. The main purpose is to study the degree to which the simple LP model embraces all of the company activities as reflected in the accounting system. The LP model itself is used as a point of departure to study the information requirements. In the next chapter, we will reverse the approach and start from a conventional accounting system and study its integration with a LP model formulation.

The main issue of this integration is to determine the information requirements that a LP model imposes on the accounting system for planning and controlling. The LP model can through its optimization capabilities be used as a planning device. On the other hand, accounting can be used to communicate and control plans and performance.

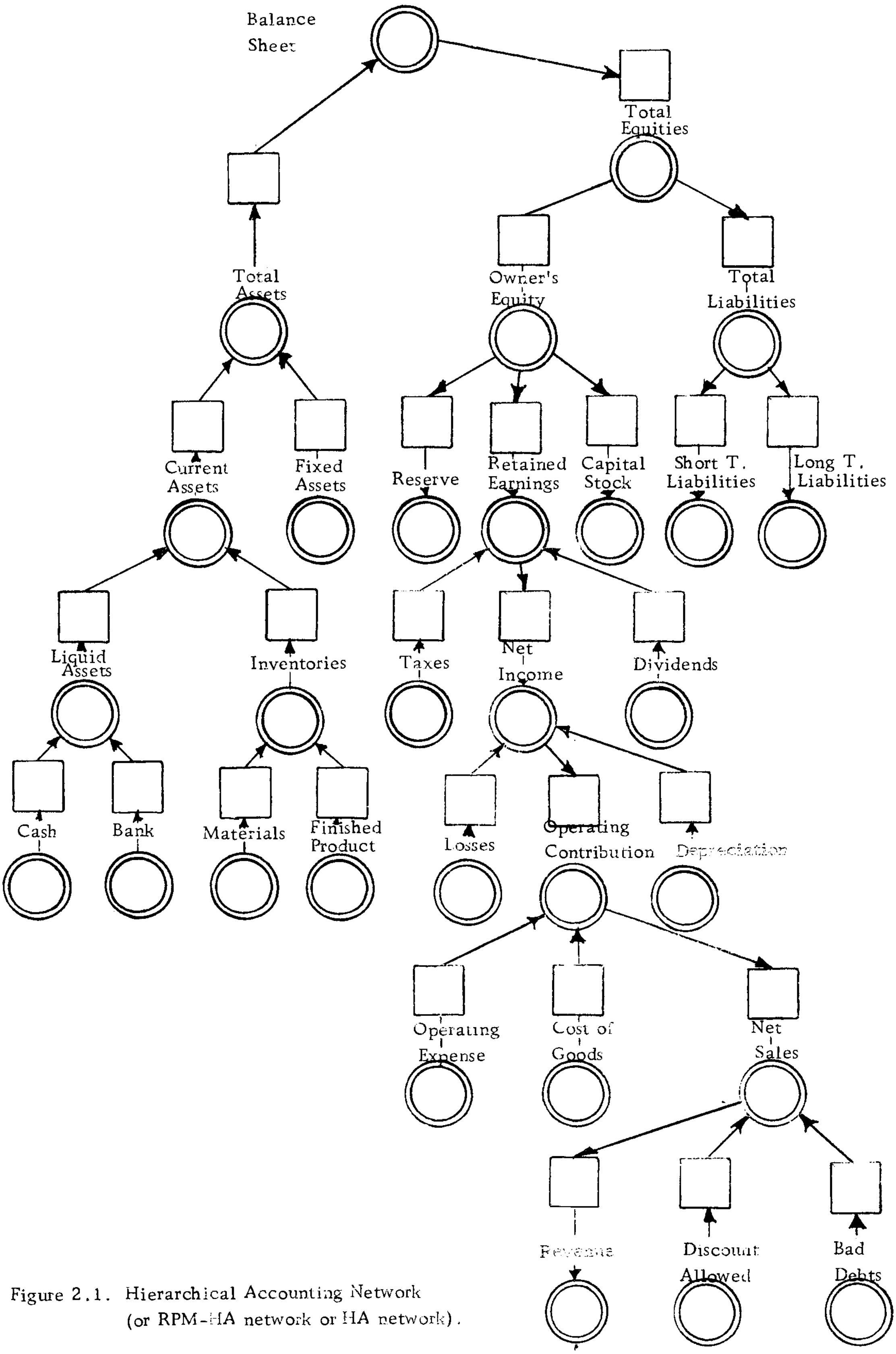


Figure 2.1. Hierarchical Accounting Network
(or RPM-HA network or HA network).

Example One (Taha, 1971)

A factory manufactures a product with each complete unit consisting of four units of component A and three units of component B. The two components (A and B) are manufactured from two different raw materials of which 100 units and 200 units are available. Three departments are engaged in the production process with each department using a different method for manufacturing the components. The following table gives the raw material per production run and the resulting units of each component:

Department	Input per run (in units)		Output per run (Units)	
	Raw Mat. 1	Raw Mat. 2	Comp. A	Comp. B
1	8	6	7	5
2	5	9	6	9
3	3	8	8	4

The objective is to determine the number of production runs for each department which will maximize the total number of complete units of the final product.

The RPM-LP network is shown in Figure 2.2. The node and variable names are:

y_1 = RM1 = units of raw material 1.

y_2 = RM2 = units of raw material 2.

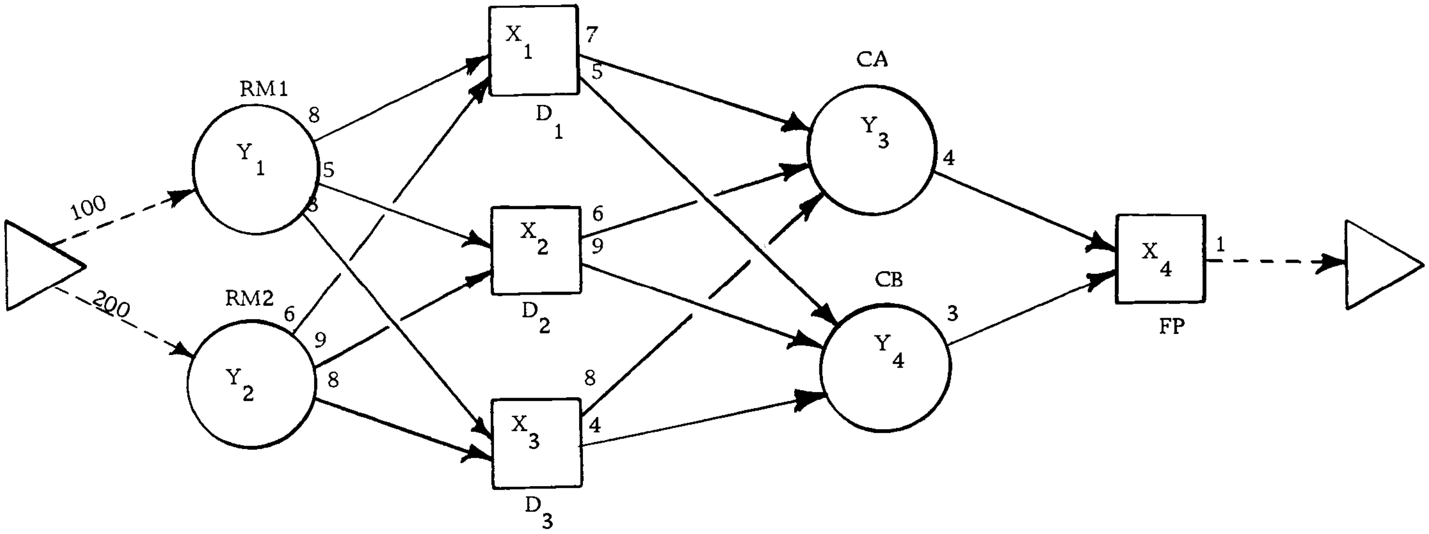


Figure 2.2. RPM-LP network for example one.

X_1 = D1 = number of production runs for department one.

X_2 = D2 = number of production runs for department two.

X_3 = D3 = number of production runs for department three.

y_3 = CA = units of component A

y_4 = CB = units of component B

X_4 = FP = units finished product

The primal model derived from Figure 2.2 is:

max. $x_0 = X_4$

subject to:

$$8X_1 + 5X_2 + 3X_3 \leq 100$$

$$6X_1 + 9X_2 + 8X_3 \leq 200$$

$$7X_1 + 6X_2 + 8X_3 - 4X_4 \geq 0$$

$$5X_1 + 9X_2 + 4X_3 - 3X_4 \geq 0$$

In order to derive the extent to which this LP model is a planning tool for the total set of activities of a firm, we shall construct a RPM-HA network above the LP network and stop the construction of this tree when a root is reached. Figure 2.3 shows the end result of this construction process applied on example one.

The most important aspect is the coupling of the RPM-LP network with the RPM-HA network. A number of conventions are helpful:

1. Each resource node of a RPM-LP model should have a corresponding "essential account". An essential account

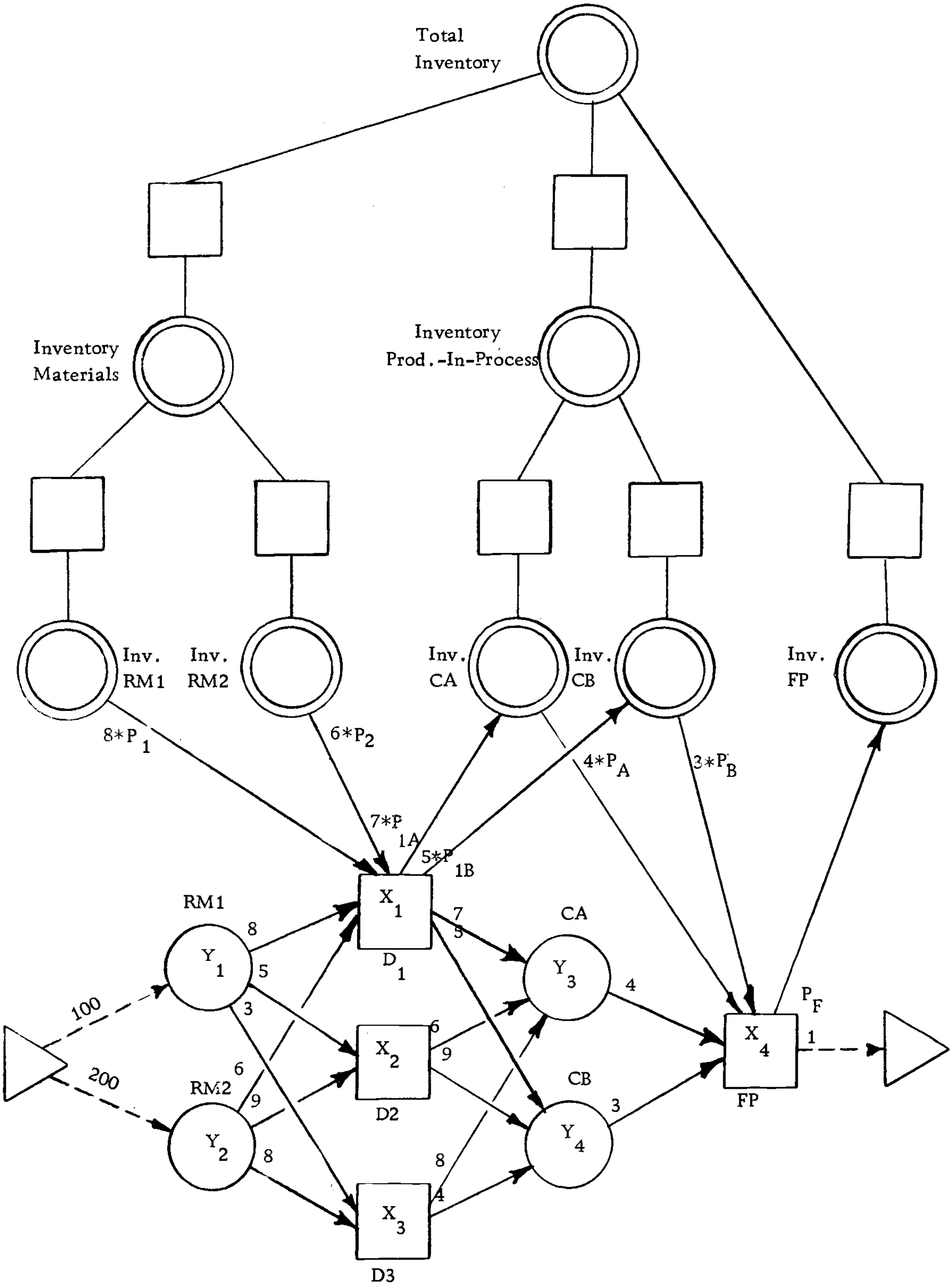


Figure 2.3. Construction of the HA-LP network to determine decision space of LP model.

can be defined as a terminal account of the RPM-HA tree. The accounts above the essential account can be called "aggregate accounts" or "higher level accounts". A one-to-one relationship exist between the essential account nodes of the HA network and the resource nodes of the LP network. For example, the raw material one resource node of the RPM-LP network has an equivalent account node "inventory raw material one". The only exception to this rule exists at the sink node; the finished product resource node of the RPM-LP network is replaced by the sink node as the objective function is centered around this node.

2. With each RPM-LP activity corresponds an accounting transaction. In RPM-LP, an activity node is an expression of a cause and effect relationship that exists among all the resource nodes it connects. An analogous cause and effect relation will activate a HA network by connecting all the essential account nodes of the HA network to the activity nodes in the same manner as the corresponding LP resource nodes are. In Figure 2. 3 these connections have been made only for X_1 and X_4 in order to keep the picture clear. Variables X_2 and X_3 should be connected in a manner analogous to X_1.

3. Because accounting conventionally uses monetary terms, the physical quantities of the activity nodes will have to be multiplied by a cost or price factor in order to initiate a correct updating of account balances. Basically, by multiplying the technological coefficients by the standard cost of resources and adding the processing cost, the monetary cost flow can be determined. However, these computations can be complex and depend also on the evaluation method adopted in the organization.

Let us assume the simple case of no conversion cost for each of the four activities. Let p_1 and p_2 represent the standard cost of materials one and two, and p_{iA} and p_{iB} (for i=1, 2, 3) the standard cost for components A and B at each of the three production centers.

Let us further assume that the cost of materials used at each of the production centers is to be divided in proportion to the units of output of components A and B. Then costs are determined as:

$$p_{1A} = p_{1B} = (8p_1 + 6p_2)/12$$

$$p_{2A} = p_{2B} = (5p_1 + 9p_2)/15$$

$$p_{3A} = p_{3B} = (3p_1 + 8p_2)/12$$

Assuming that the inventories are evaluated by averaging, and the present levels of inventory of A and B are given by Y_A and Y_B with a standard cost of c_A and c_B, then the new standard costs p_A and

p_B are computed as:

$$p_A = (c_A y_A + 7p_{1A}X_1 + 6p_{2A}X_2 + 8p_{3A}X_3)/(Y_A + 7X_1 + 6X_2 + 8X_3)$$

$$p_B = (c_B Y_B + 5p_{1B}X_1 + 9p_{2B}X_2 + 4p_{3B}X_3)/(Y_B + 5X_1 + 9X_2 + 4X_3)$$

The cost price of the finished product p_F is determined as: $p_F = 4p_A + 3p_B$.

These values can then be placed on the arrows that connect the RPM-LP network with the RPM-HA network, inorder to change monetary flows into physical flows and vice-versa. The activity nodes represent the cause and effect relationship that determines transactions to be booked in the simple accounting system. The two different types of transactions are:

a. For the production of components A and B by production center one. (Similar booking can be made for production centers 2 and 3).

Dr:	Inventory Comp. A $(7/12)(8p_1 + 6p_2)X_1$	
	Inventory Comp. B $(5/12)(8p_1 + 6p_2)X_1$	
Cr:	Inventory Mat. 1	$8p_1X_1$
	Inventory Mat. 2	$6p_2X_1$

b. For the assemblage of components A and B into the finished product:

Dr:	Inventory Fin. Prod.	$(4p_A + 3p_B)X_4$
Cr:	Inventory Comp. A	$4p_AX_4$
	Inventory Comp. B	$3p_BX_4$

By constructing the HA-LP network, it becomes possible to see the degree to which the LP model embraces all of the companies' activities. All resource nodes of the RPM-LP model have a counterpart in the accounting inventory subtree. The accounting structure drawn in Figure 2.3 is only part of a total RPM-HA network. This model limits itself to the optimal production of inventories.

III. THE CONSTRUCTION OF THE RPM-LP MODEL WITH MONETARY AND PHYSICAL FLOWS

In Chapter II it was demonstrated that a cause and effect relationship is fundamental to the interfacing of the LP model and the HA model. In this chapter we shall directly study the cause and effect relationship in an accounting information system through the use of an Interaccount Flow Network (or IAF network).

First, the spread sheet matrix and chart will be introduced. Second, an equivalent network, the RPM Interaccount Flow Network (RPM-IAF), will be derived from the spread sheet matrix. This RPM-IAF network will have two further developments. The last section of this chapter will show how the RPM-IAF network can be used to assist in the formulation of an RPM-LP model that involves monetary as well as physical flows. Later, in Chapter IV, the relationship between the RPM-IAF network and the incidence matrix will be developed. This latter matrix can be used as a means for updating the account balances of the accounting system. To conclude, Figure 3.5 gives a summary picture of the relationship among all the networks and matrices of Chapters II, III and IV.

The Spread Sheet Analysis

The total value of a firm can be categorized from two different viewpoints: (1) by the type of assets, which indicate how the sources are invested; (2) by the type of claimants that have a claim on the total value of the firms' sources. Let $Y = \{ Y_A, Y_E\} = \{ Y_{a_1}, Y_{a_2}, \ldots Y_{a_m}, Y_{e_1}, Y_{e_2}, \ldots Y_{e_n} \}$ be an ordered column vector of m asset and n equity account balances of an accounting system. Suppose that the balances of equity accounts have their values made negative, while the asset account balances are positive. Then the fundamental double entry accounting equation can be expressed as $e*Y=O$, where e is a unit row vector of order m+n.

Further, let

$$\Delta Y = Y_2 - Y_1 \tag{3.1}$$

be the difference between the ending account balances (Y_2) and the beginning balances (Y_1), and let s_{ij} be the amount for which account i is credited and account j is debited during the period between the calculation of the beginning and ending account balances. The <u>spread sheet matrix</u> (S) (Ijiri, 1965) is then defined as the square matrix $[s_{ij}]$ of order $(m+n)*(m+n)$:

$$S = \begin{vmatrix} s_{11} & s_{12} & \cdots & s_{1,m+n} \\ s_{21} & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ & & & \cdot \\ s_{m+n,1} & \cdots & & s_{m+n,m+n} \end{vmatrix}$$

The sum of s_{ij} over the columns j ($s_{i.}$) represent the total of the credit entries for account i during a given period, while $s_{.j}$ is the total of the debit entries for account j during the same period. With respect to the matrix S, $s_{i.}$ is a column vector and $s_{.j}$ a row vector. The difference between ending and beginning balances can then be expressed in terms of the transaction quantities:

$$\Delta Y = s'_{.j} - s_{i.} \qquad i = j = 1, \ldots m+n$$

or

$$= (S' - S) * e \qquad (3.2)$$

where S' is the transpose of the spread sheet matrix S and e is a m+n unit column vector. Then equation (3. 1) can be restated as

$$Y_2 = Y_1 + (S' - S) * e \qquad (3.3)$$

It can be commented that our initial assumption about negative balance values for equity accounts is respected in equation (3. 3) because $S*e = \{ s_{i.} \}$ is subtracted from Y_1. Also, it will be assumed that the diagonal elements of S are zero, as it is senseless to assume the same debit and credit booking within one account. As usual the profit and loss account is considered an equity account, whose balance, however, can be positive or negative.

The spread sheet chart (Ijiri, 1965) is nothing more than an expansion of the spread sheet matrix with the purpose of incorporating equation (3. 3) in a format that will facilitate the derivation of the financial statements. Table 3. 1 shows the basic scheme, in which $Y^-_{a_i}$ $(Y^+_{a_i})$ refers to the credit (debit) total of account i.

Table 3. 1 The spread sheet chart.

	BB[1]	1	2		m	m+1		n	Y^-	EB[1]
BB		Y_{1a_1}	Y_{1a_2}	...	Y_{1a_m}	0	. . .	0		
1	0	s_{11}	s_{12}	...	$s_{1,m}$	$s_{1,m+1}$	...	$s_{1,m+n}$	$Y^-_{a_1}$	0
2	.	s_{21}	s_{22}	...	$s_{2,m}$	$s_{2,m+1}$	...	$s_{2,m+n}$	$Y^-_{a_2}$	0
.	.	.			.	.		.	.	.
.	.	.			.	.		.	.	.
.	.	.			.	.		.	.	.
m	0	s_{m1}		...	$s_{m,m}$	$s_{m,m+1}$	...	$s_{m,m+n}$	$Y^-_{a_m}$	0
m+1	Y_{1e_1}	$s_{m+1,1}$		...	$s_{m+1,m}$	$s_{m+1,m+1}$		$s_{m+1,m+n}$	$Y^-_{e_1}$	Y_{2e_1}
.	.									
.	.									
.	.									
m+n	Y_{1e_n}	$s_{m+n,1}$		...	$s_{m+n,m}$	$s_{m+n,m+1}$		$s_{m+n,m+n}$	$Y^-_{e_n}$	Y_{e_n}
Y^+		$Y^+_{a_1}$			$Y^+_{a_m}$	$Y^+_{e_1}$	. . .		$Y^+_{e_n}$	
EB		Y_{2a_1}			Y_{2a_m}	0	. . .		0	

[1]BB = Beginning Balance EB = Ending Balance

The following example illustrates how the spread sheet framework can be used for the derivation of the three most important financial statements: income statement, balance sheet, and the funds flow statement. Equations (3.2) and (3.3) are verified in this example.

Example Two

The beginning balance sheet of a company is given in Table 3.2, and the transactions of the period are given in Table 3.3.

Table 3.2. The beginning-of-period balance sheet.

Cash (C)	5,000	Accounts Payable (P)	18,000
Receivables (R)	20,000	Long Term Liab. (L)	7,500
Raw Materials (M)	5,000	Equity (E)	17,000
Finished Product (F)	2,500		
Fixed Assets (A)	10,000		
	42,500		42,500

The use of abbreviated symbols will be useful in filling in the spread sheet chart. For example, the transaction s_{CM} will refer to the payment in cash for the purchase of raw material, and is entered in the cash row and the raw material column.

Table 3. 3. Transaction Data.

1. Purchase of raw material for $17, 500 ($7, 875 paid in cash and $9, 625 bought on credit).

Dr:	Inventory Raw Mat. (M)	17, 500	or s_{CM} = 7, 875
Cr:	Cash (C)	7, 875	s_{PM} = 9, 625
	Acc. Payable (P)	9, 625	

2. Production consumes for $19, 500 raw material

Dr:	Inventory Finished Prod. (F)	19, 500	or s_{MF} = 19, 500
Cr:	Inventory Raw Material (M)	19, 500	

3. Variable processing cost: $7, 675

Dr:	Finished Product (F)	7, 675	or s_{CF} = 7, 675
Cr:	Cash	7, 675	

4. Fixed cost of the period: $2, 700

Dr:	Income (I)	2, 700	or s_{PI} = 0. 500
Cr:	Accounts Payable (P)	2, 700	

5. Collection of Receivables: $17, 925

Dr:	Cash (C)	17, 925	or s_{RC} = 17, 925
Cr:	Accounts Receivables (R)	17, 925	

6. Payments for short term liabilities: $12, 325

Dr:	Accounts Payable (P)	12, 325	or S_{CP} = 12, 325
Cr:	Cash (C)	12, 325	

Table 3. 3 (continued)

7.	Sales for \$29, 875; cost of goods \$27, 175; 11, 925 sold on credit.			
	Dr: Sales (S)	29, 875	or s_{FS}	= 27, 175
	Cr: Inv. Finished Prod. (F)	27, 175	s_{IS}	= 2, 700
	Income (I)	2, 700		
	Dr: Cash (C)	11, 950	or s_{SC}	= 11, 950
	Accounts Receivable (R)	17, 925	s_{SR}	= 17, 925
	Cr: Sales (S)	29, 875		
8.	Depreciation fixed assets: \$1, 500			
	Dr: Income (I)	1, 500	or s_{AI}	= 1, 500
	Cr: Fixed Assets (A)	1, 500		
9.	Investment in new assets through a long term loan.			
	Dr: Fixed Assets (A)	3, 000	or s_{LA}	= 3, 000
	Cr: Long Term Liab. (L)	3, 000		

Formulas (3.2) and (3.3) provide the same result in absolute value as those directly derived through the spread sheet chart (Table 3.4):

$Y = (S'-S)*e$

$$\Delta Y = \begin{vmatrix} 0 & 17,925 & -7,875 & -7,675 & 0 & 11,950 & -12,325 & 0 & 0 \\ -17,925 & 0 & 0 & 0 & 0 & 17,925 & 0 & 0 & 0 \\ 7,875 & 0 & 0 & -19,500 & 0 & 0 & 9,625 & 0 & 0 \\ 7,675 & 0 & -19,500 & 0 & 0 & -27,175 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.0 & -1,500 \\ -11,950 & -17,925 & 0 & 27,175 & 0 & 0 & 0 & 0 & 2,700 \\ 12,325 & 0 & 9,625 & 0 & 0 & 0 & 0 & 0 & -500 \\ 0 & 0 & 0 & 0 & -3.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1,500 & -2,700 & 500 & 0 & 0 \end{vmatrix} * \begin{vmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{vmatrix} = \begin{vmatrix} 2,000 \\ 0 \\ -2,000 \\ 0 \\ 1,500 \\ 0 \\ 2,200 \\ -3,000 \\ 700 \end{vmatrix}$$

$Y_2 = Y_1 + \Delta Y$

$$Y_2 = \begin{vmatrix} 5,000 \\ 20,000 \\ 5,000 \\ 2,500 \\ 10,000 \\ 0 \\ -18,000 \\ -7,500 \\ -17,000 \end{vmatrix} + \begin{vmatrix} 2,000 \\ 0 \\ -2,000 \\ 0 \\ 1,500 \\ 0 \\ 2,200 \\ -3,000 \\ -700 \end{vmatrix} = \begin{vmatrix} 7,000 \\ 20,000 \\ 3,000 \\ 2,500 \\ 11,500 \\ 0 \\ -15,800 \\ -10,500 \\ -17,700 \end{vmatrix}$$

The spread sheet chart is illustrated in Table 3.4, and the derived financial statements in Table 3.5.

The Interaccount Flow Network

It is possible to construct a RPM network that is equivalent to the spread sheet chart of Table 3.1. The following basic conventions are used:

1^o With each transaction element s_{ij} corresponds an activity node (a square); s_{ij} was earlier defined as a simple transaction for which account i receives a credit and account j a debit for the given value s_{ij}. In this context, the activity node will also be called a simple transaction node.

2^o With each account of the rows (or the columns) corresponds a resource node (a circle). This node represents an account balance and hence will be called an account node.

3^o The direction of the arrows around simple transaction (ST) nodes is such that the ST node connects acount i with account node j. With respect to the account node, a leaving arrow refers to the crediting of an account, while an entering arrow indicates the debiting of an account.

Table 3.4. Spread Sheet Chart for example two.

		C	R	M	F	A	S	P	L	I	Cr. Tot	EB
	BB	5,000	20,000	5,000	2,500	10,000	-	-	-	-	42,500	-
Cash (C)	-			7,875	7,675			12,325			27,875	-
Receiv. (R)		17,925									17,925	-
Raw Mat. (M)					19,500						19,500	-
Fin. Pr. (F)							27,175				27,175	-
Fix. As. (A)	-									1,500	1,500	-
Sales (S)	0	11,950	17,925								29,875	0
S. T. Liab. (P)	18,000			9,625						500	28,125	15,800
L. T. Liab. (L)	7,500					3,000					10,500	10,500
Income (I)	17,000						2,700				19,700	17,700
Dr. Total	42,500	34,875	37,925	22,500	29,675	13,000	29,875	12,325	0	2,000	176,875	
Ending Asset Balance		7,000	20,000	3,000	2,500	11,500	-	-	-	-		44,000

Table 3.5. Financial Statements for example two.

The end-of-period Balance Sheet

Cash (C)	7,000	Accounts Payable (P)	15,800
Receivables (R)	20,000	Long Term Liab. (L)	10,500
Raw Materials (M)	3,000	Equity (E)	17,700
Finished Prod. (F)	2,500		
Fixed Assets (A)	11,500		
	44,000		44,000

The Income Statement

Sales	29,875
Cost Finished Product Sold	27,175
Gross Profit	2,700
Fixed Cost	500
Depreciation	1,500
Net Income	700

The Statement of Funds Flow

Income from Operations:		
Net Income	700	
Depreciation	1,500	
Gross Income		2,200
Income from other sources:		3,000
Total Sources		5,200
Application of Funds		3,000
Increase in Working Capital:		2,200

4° The flow at each account node should be an equality type flow.

A network obtained according to these rules is shown in Figure 3.1 and will be called the interaccount flow network (or IAF network).[3]

The following observations serve to link traditional accounting thinking to the network approach:

- Previously, we indicated that the diagonal elements of the spread sheet matrix should equal zero. In RPM terms, a positive diagonal value would correspond to a simple transaction node that is connected on both sides to the same account node, and thus would be irrelevant to the value of the account balance.
- It is helpful to think of fixed assets as prepaid or stored costs (Horngren, 1965), and of depreciation as a gradual flow from the asset value into the product cost. By the same token, the income from operations can be defined as a gradual increase in the total networth.
- There are several ways in which income of operations could be determined. For example, the variable cost could be connected directly to the income account instead

---

[3] A similar network has been proposed by Charnes, Cooper, and Ijiri (1963). The nodes of their graph correspond to account balances, while the arrows represent the flow of transaction amounts.

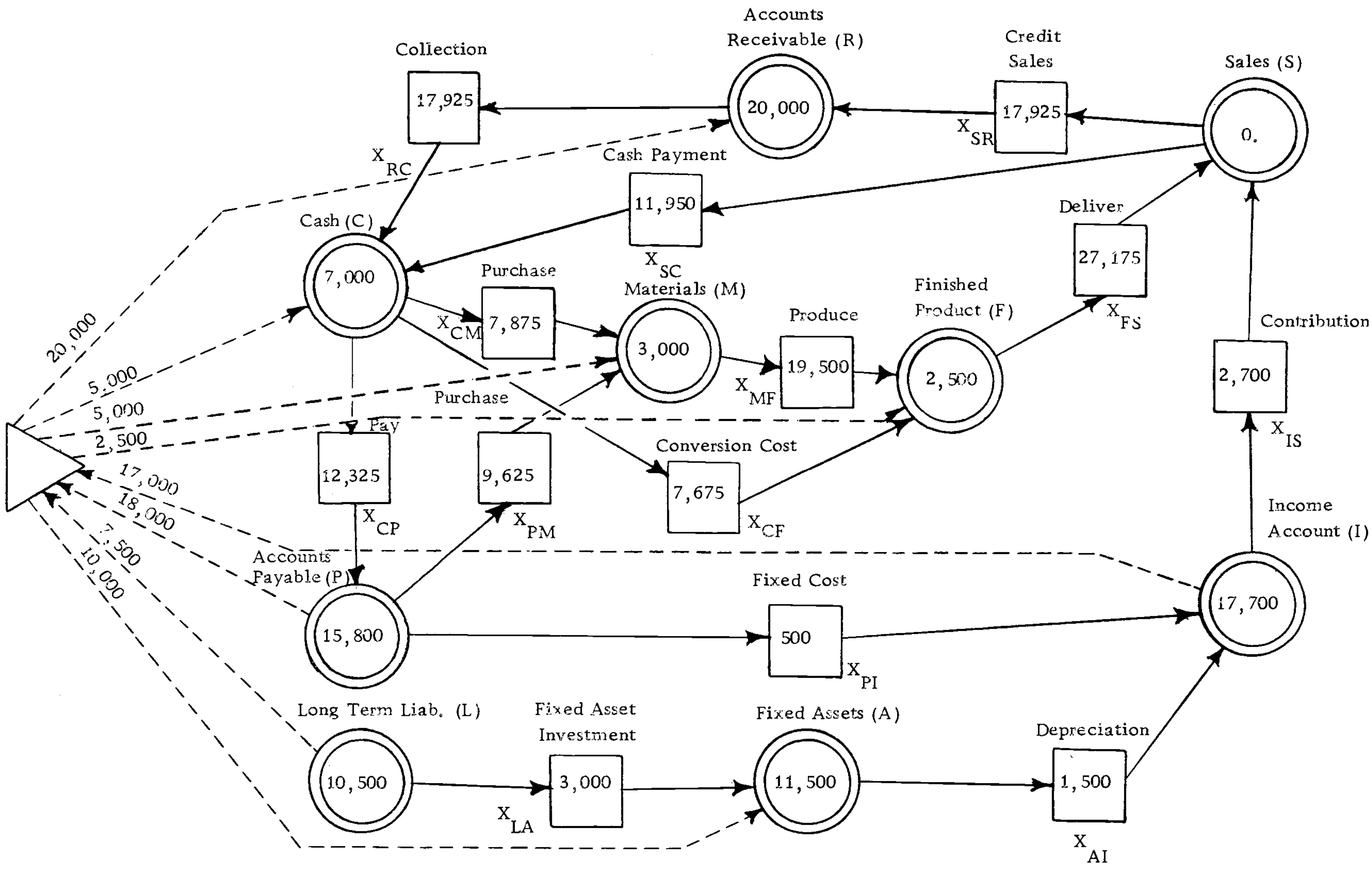


Figure 3. 1. The Interaccount Flow Network (RPM-IAF).

of the sales account. In Figure 3.1 the sales account acts more or less as a profit and cost account.

- It is also clear that the variable and fixed cost flows can be considered as aggregations of more detailed transactions. These aggregations are obtained through the end-of-period closing of intermediate accounts. The closing transactions could be included in the IAF network by incorporating more intermediate ST and account nodes before the aggregating ST node.
- Whether these flows represent accrual or cash flows depends on whether or not adjustments have been made for accruals.
- The IAF network shows how the accounting system represents a closed system of accounts and monetary flows. The implication is that the system of equality constraints derived from the network by applying the RPM rules, must have one dependent equation. Indeed, there is still the equation expressing the equality between the total of debit entries and the total of credit entries.

At this point, a number of assumptions limit the use of the IAF network:

1. Each ST node has only one entering and one leaving arrow, because the elements of the spread sheet matrix represent

simple transactions. On the other hand, the account nodes will have as many entering (or leaving) arrows as there are nonzero elements in the corresponding column (or row) of the spread sheet matrix.

2. Each ST node represents a positive and monetary flow, due to the separation of the debit and credit amounts booked to each account.

3. The coefficients for the arrows are all equal to one, because of the IAF network construction rules one and two.

In the next section, we shall study how these limitations will be eliminated in the transformation process from the RPM-IAF network to the RPM-LP network.

The RPM-LP Decision Model

A first step in securing the derivation of the RPM-LP network for a firm's operating subsystem is to assure that the proper accounts are incorporated in the network. Basically, two conditions have to be satisfied: a linearity requirement must be met, and the accounts must pertain to the operations aspects of the firm (i. e., they should belong to the working capital segment[4] of the accounting

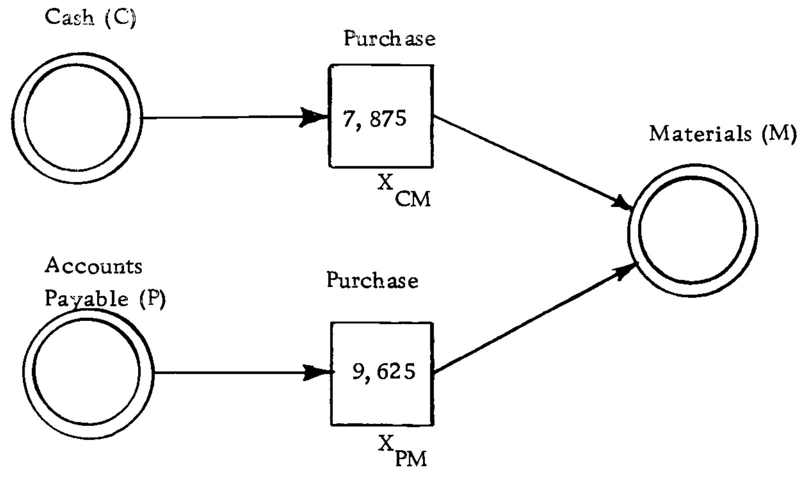
[4] This concept is also used in the conceptual framework of the Funds Flow Statement (see e. g., Horngren, 1965).

system). Very frequently both of these conditions cover the same subset of accounts. This means that we are interested in modeling changes in current assets and liabilities (the so-called working capital segment) accounts and transactions in so far they satisfy the linearity assumptions or can be approached in a piecewise linear fashion. Changes in the non-working capital such as losses, capital investments etc.... are changes in monetary value of assets and equities by actions that are not considered to be initiated by the set of current repetitive decisions of the operating subsystem.

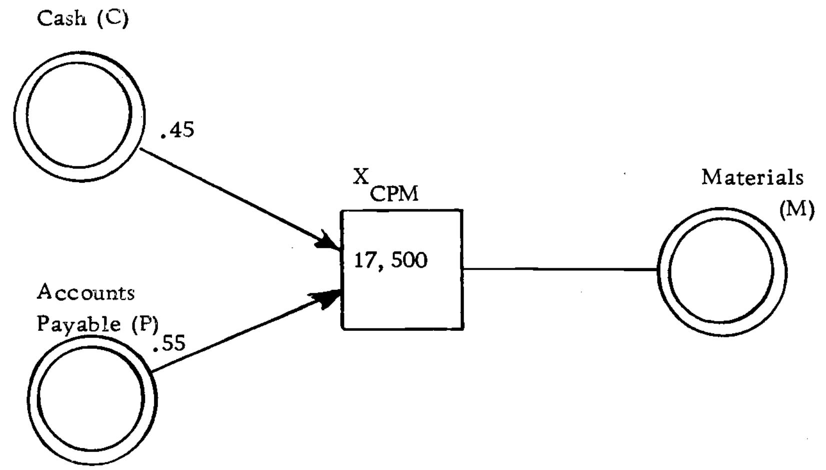
A second step is to transfrom the reduced IAF network to the desired RPM-LP format. So far, the partial IAF network resembles a monetary process flow, useful for product costing, budgeting and simulation purposes, but not yet for optimization purposes. A first reason stems from assumption one underlying the IAF network. Each transaction node corresponds to a simple accounting transaction; however, a conventional LP model typically has several entering and leaving arrows for each activity node in order to reduce the redundancy in the set of constraints to be derived from the network. A second reason is, that the IAF is not activity centered; it is centered around the account nodes and only uses activity nodes for showing their transactions. The physical flows, capacity constraints, policy restrictions and certain equality constraints on activities have not been added. Thirdly, one of the RPM conventions for the LP models is that a circle corresponds to a constraint; but in Figure 3.1 each account node does not necessarily qualify as a restriction, even if a policy statement is made about the required levels for each account balance. These three observations form the core of the problem associated with the derivation of a RPM-LP network from a IAF network, and will now be further discussed and illustrated.

Deriving Compound Transaction Nodes

Assuming that the value of the transactions for the period show steady state relationships among each other, then certain ST nodes can be combined into compound transaction (or CT) nodes with technological coefficients that show the relative contribution from (or relative output to) their related account nodes. In example two this can be applied to the purchasing of raw material and the payment of sales. For example, the purchasing of raw materials by the use of cash payments causes a transaction of 7,875 thousands of dollars, while the purchasing on credit accumulates to 9,625 thousands of dollars. Assuming a stationary distribution between these two simple transactions, it is possible to compute the coefficient 7.875/(7.875 + 9.625) = .45 for the cash contribution and 9.625/(7.875 + 9,625) = .55 for the input of the accounts payable in the acquisition of one dollar of raw materials. See Figure 3.2 for the transformation in the network.



a. two simple transactions



b. compound transaction

Figure 3.2. Derivation of a compound transaction from two simple transactions.

This method of combining simple transactions, by which the resultant coefficients (a_{ij}) add up to one, is only applicable to monetary flows. If the technological coefficient (a_{ij}) were expressed in their physical dimensions, there is no reason why these a_{ij} would add up to one (Hadley, 1962).

Generating Physical Flow from Monetary Flow

The monetary flows of the spread sheet chart are determined by two factors: a price factor and a quantity factor. Any deviation from standards in one or in both of these factors causes a deviation in the standard monetary flow. In order to study the sensitivity of changes in these price and quantity standards, it is desirable to separate the physical and monetary flows of a model.

In practice, it may be desirable to detail the physical flow more than the monetary one, and to obtain only pertinent monetary flows by multiplying the physical quantities by cost or price factors. Example two can be extended to illustrate the separation of physical from monetary flows. Assume that the raw material purchased can be processed on either one or two machines of the plant. Machine two is a packing machine and machine one is a processing machine. The bulk raw material can be sold for retail by packing it, or this material can be first processed on machine one before being packed by machine two. The physical standard data and capacities are summarized in Table 3. 6.

Table 3.6. Physical data for example two.

	Processing	Packing
Product 1	-	2 days/ 1, 000 units
Product 2	3 days/1, 000 units	2 days/1, 000 units
Capacity	10.5 days	22 days

The standard cost of raw materials is determined at $2 per unit and the standard price of the two products is obtained by adding a contribution of 10% to each product cost. Also given is the processing cost of $.80 per unit on machine one and $.50 per unit on machine two. Assume also a minimum demand of 6, 000 units for product one and 3, 000 units for product two.

In order to keep an exact relationship between the physical and monetary flows, the cost factors of one unit product have to be accumulated over the several processes.[5] The cost for product one, that goes only through the packing process, is then 2 + .5 = 2.5 dollars, while product two has a variable cost of 2 + .80 + .50 = 3.30 dollars.

---

[5] It should be noted that the RPM network of physical flows is very useful in organizing the computations for complex product costing problems. Basically, product cost accounting requires the accumulation of the variable manufacturing costs, caused by the different production processes and material uses and losses, for a given period of time.

Figure 3. 3 shows the new constraints that have been added to the original IAF network. The shaded nodes represent physical flows. Besides the capacity constraint, equality constraints are used to relate physical to monetary flows: constraint Y_8 relates the production activities (X_1 and X_2) to the conversion cost (X_{CF}), and constraint Y_9 defines the contribution.

Account Balance Constraints

Because a firm is to be considered as an on-going concern, management wants to maintain certain levels of asset and equity accounts. Restrictions and policy statements can be made for each individual accoumt, or a general relationship between the beginning and ending balance can be made. To illustrate this on example two, we assume that management stated, with respect to expectations for the next period, that all end-of-period balances should equal the beginning balance, except that cash should be $2, 000 more and the raw material account $1, 000 less than their respective beginning-of-period balances.

An important shortcoming of the IAF network and its further extensions, as shown in Figure 3. 3, is that it does not give the inter-period constraints. For example, these networks do not reflect the fact that the availability of raw materials might be a constraint on the production output. Indeed, sales are always paid in cash that makes

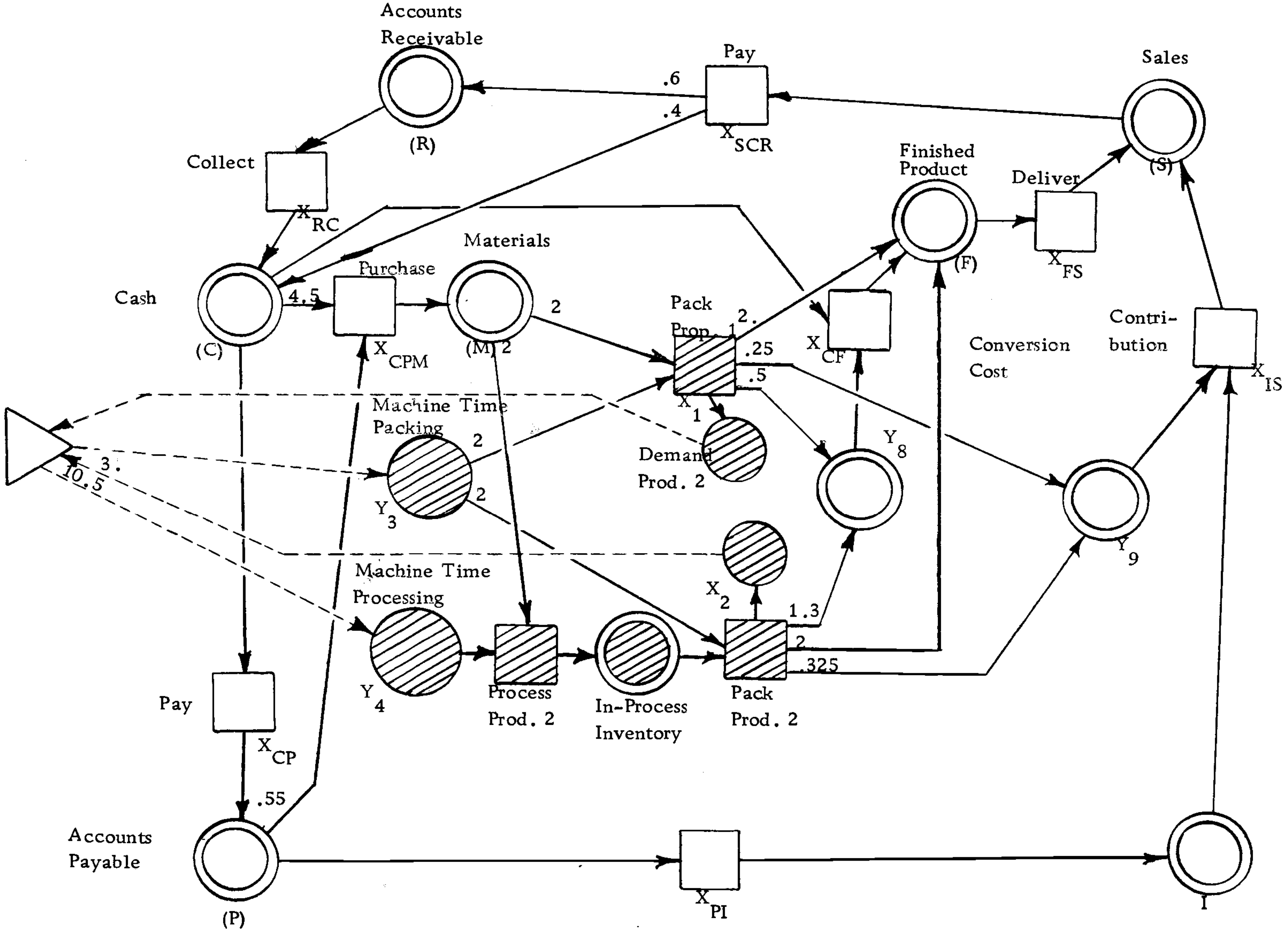


Figure 3.3. The IAF network with compound transactions and physical flows.

money immediately available for the acquisition of new raw materials within the same period. However, the intent of this illustration is to limit the purchases to $17, 500 due to storage capacity. Similarly, the collection of accounts receivable cannot exceed its beginning balance, while payments for accounts payable are assumed to be no bigger than the beginning-of-period balance of the accounts payable. Such constraints, expressing interperiod and other logical relationships among money transfers have to be investigated, added to the network, and can possibly make some of the account node relationship redundant.

In general, the account node relationships can be converted into LP constraints by considering them as the beginning-of-the period accounting balance. Then the desired ending balance for an account should be set equal to the beginning balance plus the total of all transactions on the entering arrows minus the total of all transactions for the leaving arrows. We will now formulate these account balance restrictions for the new information on example two.

1. Raw material related constraints:

   a. The purchase capacity constraint is:

$$X_{CPM} \leq 17.500.$$

   b. The raw materials balance should be $2, 000 less than before; let M_E be the ending materials balance, and M_B the beginning balance, then:

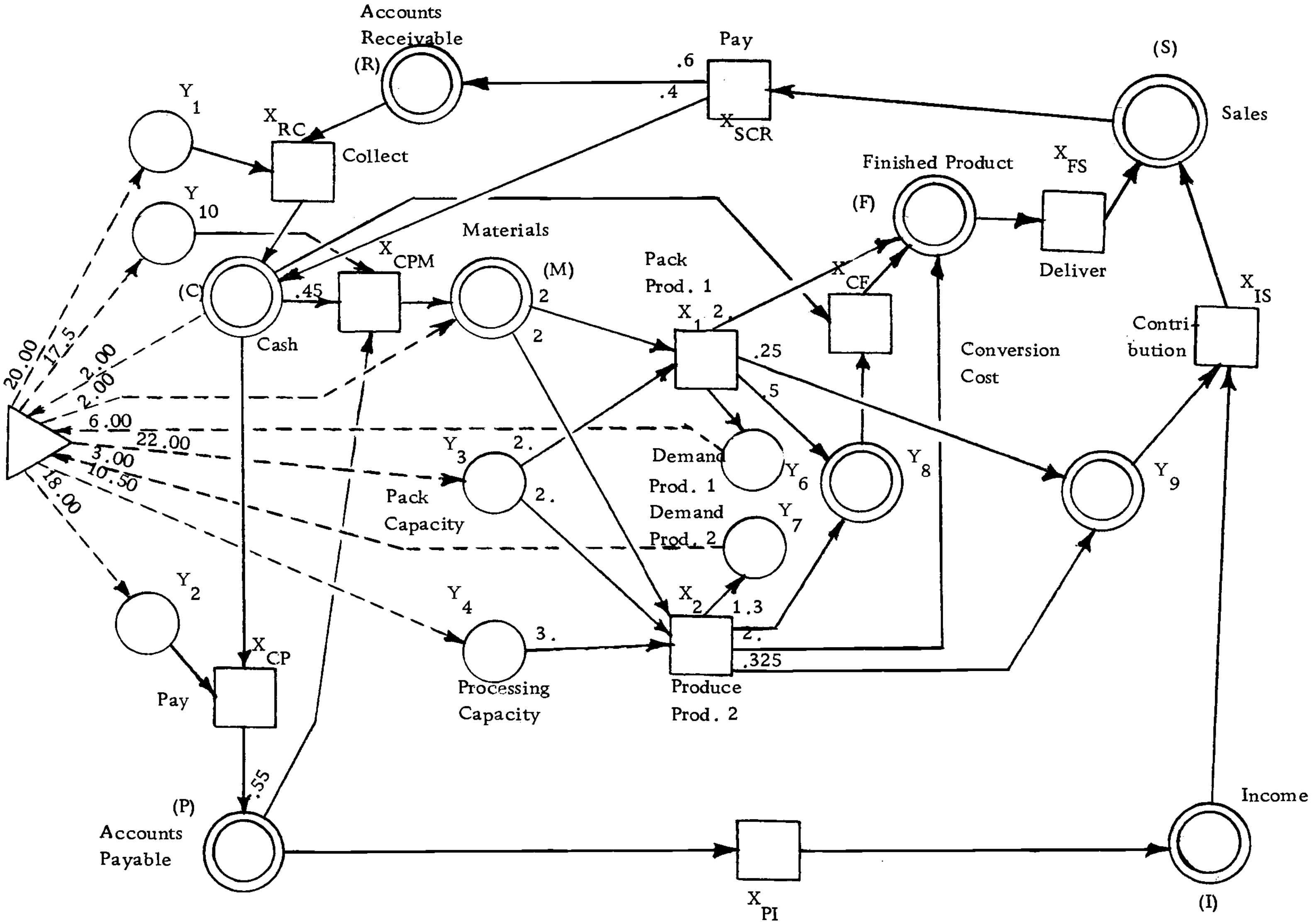


Figure 3.4. RPM network for restrictions on the LP model of example two.

$$M_E = M_B - 2.000 = M_B + X_{CMP} - 2\ (X_1 + X_2),$$

$$\text{or } 2X_1 + 2X_2 = X_{CMP} = 2.000.$$

2. Accounts Payable and Receivables Constraints:

a. Let C_B be the beginning cash balance and R_B the beginning accounts receivable balance, then:

$$\text{at } y_2 : \quad X_{CP} \leq C_B \quad \text{or } X_{CP} \leq 18.000;$$

$$\text{at } y_1 : \quad X_{RC} \leq R_B \quad \text{or } X_{RC} \leq 20.000.$$

3. Cash Constraints:

a. Let C_E be the end of period cash balance and C_B the beginning balance, then:

$$C_E = C_B + 2.000 = C_B + X_{RC} + .4\ X_{SRC} - X_{CF}$$

$$- .45\ X_{CPM} - X_{CP};$$

$$\text{or } X_{RC} + .4\ X_{SRC} - X_{CF} - .45\ X_{CPM} - X_{CP} = 2.000.$$

The formulation of the other constraints is readily derived from the network. Figure 3.4 gives the complete RPM network for the total set of restrictions.

Table 3.7 contains a list of the restrictions in equation form. Most account constraints are not connected to the source, because the coefficient on their arrow to the source equals zero. Note also that the processing and packing activities for product two have been reduced to one variable (X_2), in order to reduce redundancy.

Table 3.7. Restrictions for the LP model of example two.

at Y_1	X_{RC}	$\leq 20.$
at Y_{10}	X_{CMP}	≤ 17.5
at Y_2	X_{CP}	$\leq 18.$
at Y_3	$2X_1 + 2\,X_2$	$\leq 22.$
at Y_4	$3\,X_2$	≤ 10.5
at Y_6	$-X_1$	$\leq -6.$
at Y_7	$-X_2$	$\leq -3.$
at C	$X_{RC} - X_{CP} + .4\,X_{SCR} - X_{CF} - .45\,X_{CMP}$	$= 2.$
at M	$X_{CMP} \quad -2X_1 - 2\,X_2$	$= -2.$
at P	$X_{CP} - .55\,X_{CMP} - X_{PI}$	$= 0$
at R	$-X_{RC} + .6\,X_{SCR}$	$= 0$
at Y_8	$-X_{CF} + .5\,X_1 + 1.3\,X_2$	$= 0$
at Y_9	$.25\,X_1 + .325\,X_2 - X_{IS}$	$= 0$
at F	$X_{CF} \quad 2.\,X_1 + 2.\,X_2 \quad -X_{FS}$	$= 0$
at S	$-1.\,X_{SCR} \quad +X_{IS} + X_{FS}$	$= 0$

It now becomes possible to introduce an objective function, say by optimizing one of the equality constraints, or a deviation from an inequality constraint. For example, let us take the contribution constraint at node Y_9 and maximize $X_{IS} = .25 X_1 + .325 X_2$; we then obtain the conventional optimization problem, that optimizes the product contribution. It should also be noted that in this network the fixed cost X_{PI} is set equal to X_{IS} (the contribution) through the income constraint (node (I)). Indeed, X_{PI} is considered here more than just the fixed cost of operations; it forms a link between the working and non-working capital segments, and for this reason it includes whatever application of the contribution (X_{IS}) occurs, such as payment of dividends, new investments, etc.... In other words, the model breaks even with respect to income.

Similarly, other constraints can be taken out of the set of restrictions, and transformed into the objective function. If maximization of sales were the objective, we would

$$\text{maximize } X_{SCR} = X_{CS1} + 2X_1 + 2X_2 + X_{CS2} + X_{IS} \text{ or}$$

$$= X_{CS1} + 2\,x_1 + 2\,x_2 + X_{CS2} + .25\,x_1 + .325\,x_2$$

In this case, this objective would give the same results as the maximization of the contribution. However, if the objective would be to maximize the contribution and the accounts payable, in order to improve the own cash position, then the objective function becomes

$$\text{max. } M_1\,(.25X_1 + .325X_2) + M_2\,(X_{CP} - .862\,X_{CMP} - X_{IS})$$

where M_1 and M_2 are two scalars impounding the relative importance of the two subobjectives with respect to each other.

It should be noted that the constraint around the (I) node was left out of the set of constraints during the solution phase of the model, and yet, the model results in a breakeven solution with respect to income and fixed costs. As mentioned earlier, this is due to the fundamental double entry relationship, that the total of debit entries must equal the total of credit entries.

In this chapter, we studied how an accounting system can be made a part of the traditionally known production LP models, through the use of the RPM network technique. This shows how close an information system stands to a decision model. The solution of the LP model will result in a set of values that are closely related to the transaction values needed to update the accounting system. This interface aspect is studied in Chapter IV. To bring all networks and matrices of Chapters II, III and IV in proper relationship with each other, Figure 3.5 is referred to.

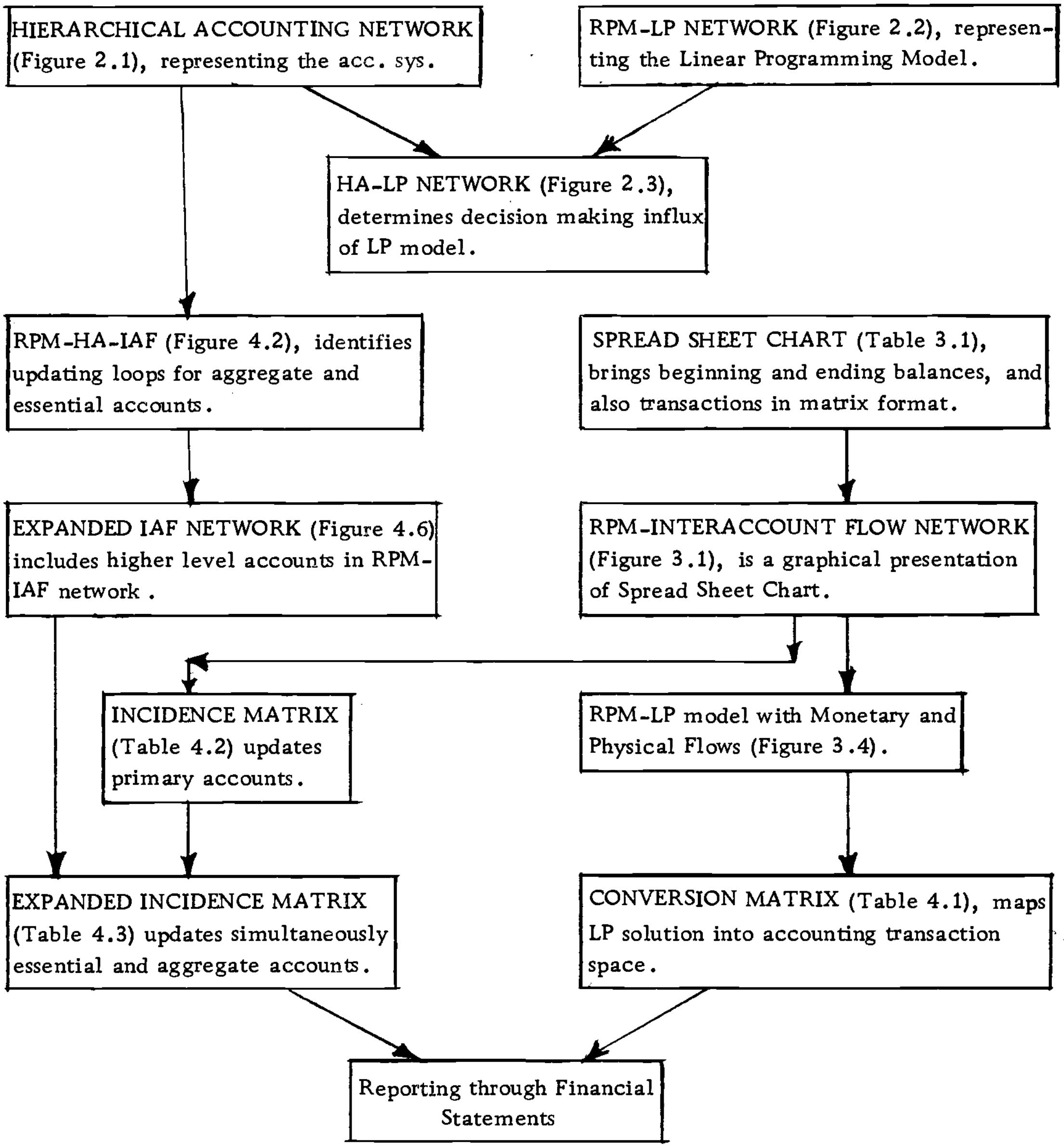


Figure 3.5. Relationships among Graphs and Matrices of Chapters II, III and IV.

IV. THE INTERFACE BETWEEN THE DECISION MODEL AND THE ACCOUNTING SYSTEM

Chapter III started with the spread sheet chart and analyzed its relationship with the LP model through the use of the RPM-IAF network. Now it will be shown how the results of the LP model can be mapped into the transactions space of the accounting system, and how the set of accounts can be updated. First, the conversion matrix will be studied as a tool to facilitate the procurement of transaction accounts. Second, the incidence matrix will be derived from the RMP-IAF network and will prove useful to update the essential accounts. Finally, the IAF network will be expanded to include all higher level accounts of the accounting system. This will lead to an expanded incidence matrix that updates simultaneously the primary and higher level accounts. This latter elaboration will be a joining of the material discussed in Chapters III and IV with the hierarchical accounting network of Chapter II.

The Conversion Matrix

The conversion matrix is a matrix that maps the LP solution into the transaction space of the accounting system. This matrix multiplied by the optimal solution of the LP model determines the transaction values needed to update the account balances. Table 4. 1 illustrates its use applied on the optimal results of example two.

Basically, the conversion matrix is optional, as it is possible to determine a LP model giving a solution for which the primal variables are identical to the transaction amounts. However, its advantages outweigh the disadvantage of its use. First, the order of the variables in the LP model no longer has to be the same as the order of transaction amounts, later to be multiplied with the incidence matrix in order to update the account balances. Second, its use helps to reduce the number of activities and constraints needed in the LP model. Compound transaction amounts are easily decomposed, and the conversion of physical into monetary quantities can be achieved by the same technological coefficients used in the LP model.

To illustrate this, Figure 3.4 has been modified, so that the result of its corresponding overdetermined model (i.e., a model with more defining constraints than is required) can directly be used to update the account balances. The model in Figure 4.1 counts 3 constraints and 3 activities more than the reduced model of Figure 3.4. For each operating transaction in the IAF network (Figure 3.1) a variable has to exist in the LP model; this implies also that all lower level accounts related to these transactions will appear as restrictions in the LP model. Finally, additional constraints are introduced to keep certain variables in proper relationship with each other. The model was solved by the RPMLP program

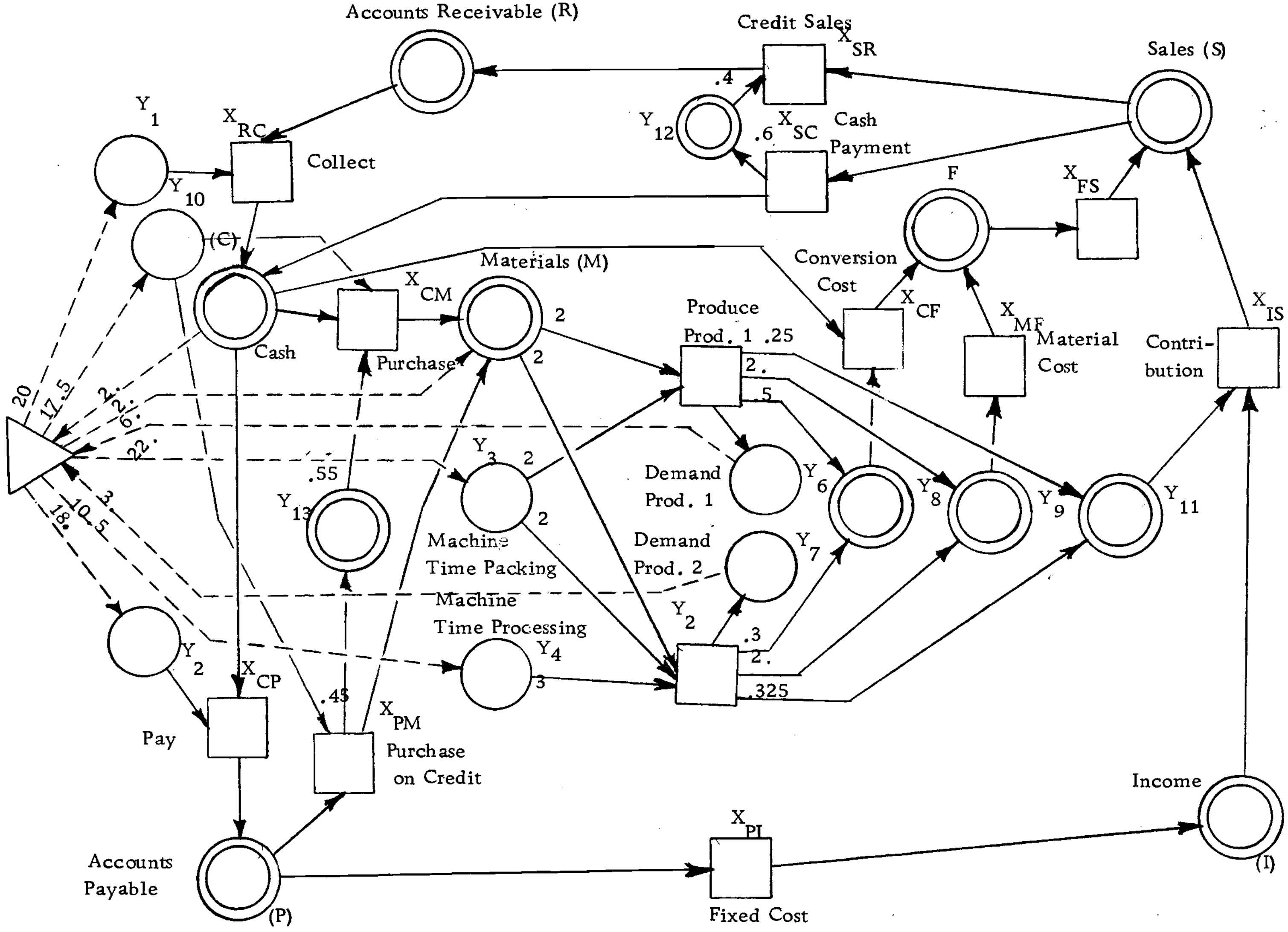
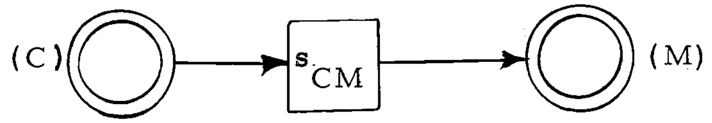


Figure 4.1. Overdetermined LP model for example two.

(see Appendix III) and gave the same solution as the one obtained after multiplying the results of the LP model by the conversion matrix. The latter is illustrated in Table 4.1.

The Incidence Matrix

In Table 3.3 we used an abbreviated symbol to represent a monetary flow of a simple transaction between two accounts. For example, $s_{CM} = r$ refers to a crediting of the cash account and a debiting of the materials account for an amount of r dollars. In network form we have:



The incidence matrix[6] (T) can then be defined as a matrix that has one column for each square and one row for each account of the IAF network. Each cell of T is then filled with a value of +1 for each entering arrow from account i into the ST node s_{ij} and a value of -1 for each leaving arrow from the ST node s_{ij} into account j. Table 4.2 illustrates the construction of an incidence matrix for the IAF network of Figure 3.1.

The order of matrix T is (m+n)*s, where m is the number of elementary asset accounts, n the number of elementary equity

[6] More properties of the incidence matrix can be found in Charnes and Cooper (1961 and 1967). Its use is also illustrated in Ijiri Y. (1965).

Table 4.1. Use of conversion matrix to determine transaction amounts from LP results.

		Conversion Matrix										LP Solution		Trans-actions
X_{CM}		0	0	0	0	.45	0	0	0	0		X_{RC} =17, 925		7, 875
X_{PM}		0	0	0	0	.55	0	0	0	0		X_{CP} = 12, 325		9, 625
X_{MF}		0	0	0	0	0	2.	2.	0	0		X_{SCR} = 29, 875		19, 500
X_{RC}	=	1.	0	0	0	0	0	0	0	0	*	X_{CF} = 7, 675	=	17, 925
X_{CP}		0	1.	0	0	0	0	0	0	0		X_{CPM} = 17, 500		12, 325
X_{FS}		0	0	0	0	0	0	0	0	1.		X_1 = 6, 250		27, 175
X_{IS}		0	0	0	0	0	0	0	1.	0		X_2 = 3, 500		2, 700
X_{SC}		0	0	.4	0	0	0	0	0	0		X_{IS} = 2, 700		11, 950
X_{SR}		0	0	.6	0	0	0	0	0	0		X_{FS} = 27, 175		17, 925
X_{CF}		0	0	0	1.	0	0	0	0	0				7, 675

Table 4.2. Incidence matrix for example two.

	X_{CM}	X_{PM}	X_{MF}	X_{RC}	X_{CP}	X_{FS}	X_{IS}	X_{SC}	X_{SR}	X_{CF}	X_{AI}	X_{PI}	X_{LA}
(C) Cash	-1			1	-1			1		-1			
(R) Receivables				-1					1				
(M) Materials	1	1	-1										
(F) Fin. Prod.			1			-1				1			
(A) Fix. Asset											-1		1
(S) Sales						1	1	-1	-1				
(P) Short T. Liab.		-1			1							-1	
(L) Long T. Liab.													-1
(I) Networth							-1				1	1	

accounts, and s the number of activity nodes in the IAF network. Because the diagonal elements of the spread sheet matrix must always equal zero, the maximum s value is (m+n)*(m+n)-(m+n) or (m+n)*(m+n-1). Each row of the incidence matrix shows which transactions or activities can cause an increase on the debit side, in case of a +1, or on the credit side, in case of a -1, of its respective account. In other words, asset account balances are increased (or decreased) for activities with a +1 (or -1) in their row, while credit account balances are increased (or decreased) for activities with a -1 (or a +1) in their rows.

If in addition to the incidence matrix we create a column vector B of the order s*1, that has the transaction amounts for each activity node of the IAF network in the same order as the columns of matrix T, then from equation (3. 1) the following relationship holds:

$$\Delta Y = (S' - S)*e = T*B \tag{4. 1}$$

The B vector is formed on the basis of the transactions given in Table 3. 4; the results for ΔY are identical to those found earlier:

$$\Delta Y = \left|\begin{array}{cccccccccc|ccc}
-1 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0
\end{array}\right| * \left|\begin{array}{c}
7.875 \\ 9.625 \\ 19.500 \\ 17.925 \\ 12.325 \\ 27.175 \\ 2.700 \\ 11.950 \\ 17.925 \\ 7.675 \\ \hline 1.500 \\ .500 \\ 3.000
\end{array}\right| = \left|\begin{array}{c}
2.000 \\ 0 \\ -2.000 \\ 0 \\ 1.500 \\ 0 \\ 2.200 \\ -3.000 \\ -\ .700
\end{array}\right|$$

In the second section of Chapter III, we studied how an IAF network could be drawn using all account balances of an accounting system and all transactions for a given period. We have shown in this section that the relationship between the incidence matrix and the IAF network is one of equivalence. Indeed, the conversion rules between them are unambiguous.

The LP model, however, does not embrace all of the decision variables of a company; at best, it will contain all of the working capital segments of the accounting system. The degree to which the LP model reflects all of the companies decisions was studied in the third section of Chapter II. Similarly, the incidence matrix corresponding to the LP model will only be a subset of the incidence matrix corresponding to the total IAF network. This means that the total incidence matrix not only allows the updating of the accounts for the optimal solution of the LP model, but also for the (non-linear) decisions that are considered here to be made externally with respect to the decision model. For instance, transactions corresponding to investment decisions, dividend payments, depreciation amounts ... can be made to correspond with a second set of columns of the incidence matrix.

Given an LP model, it is desirable to obtain all essential accounts before the degree of decision making by the LP model (i.e., level of accounts affected by the model decisions) can be determined and before the LP portion of the IAF network or the incidence matrix can be obtained. This can be accomplished by reversing the previously discussed derivation of the LP model from the IAF network. In short, because we are only interested in monetary flows, we first eliminate all physical flows and transformation coefficients, so that only monetary flows appear in the LP model. Second, account constraints or nodes should be distinguished from the other constraints. By account nodes we mean, nodes that can represent a legitimate account balance. Only activity and account nodes should be retained in the network. Third, compound activities have to be divided in simple activities, resulting in unit arrow coefficients.

The IAF network so obtained determines an incidence matrix that allows updating of accounts so that the optimal plan of the LP model can be compared with the actual results. This method allows us to find the consistency requirements to be imposed on the information system for planning and control data. However it does not mean that the minimum requirements of a total system are determined; the many data items that are needed for operations, but not for decisions are not taken into account through this method.

The Expanded Incidence Matrix

Basically, two networks of accounting have been presented sofar: the HA network in Chapter II, and the IAF network in Chapter III. These two networks can be integrated by recognizing their nature. The HA network relates the higher level accounts of the balance sheet and the income statement to the essential general ledger accounts. On the other hand, the IAF network shows how the money values are transferred among these general ledger accounts. By integrating these two networks, one simultaneously sees the value transfers among the lower and higher level accounts. Figure 4.2 shows this integration applied to example two. The lower part shows the IAF network as constructed in Chapter III.

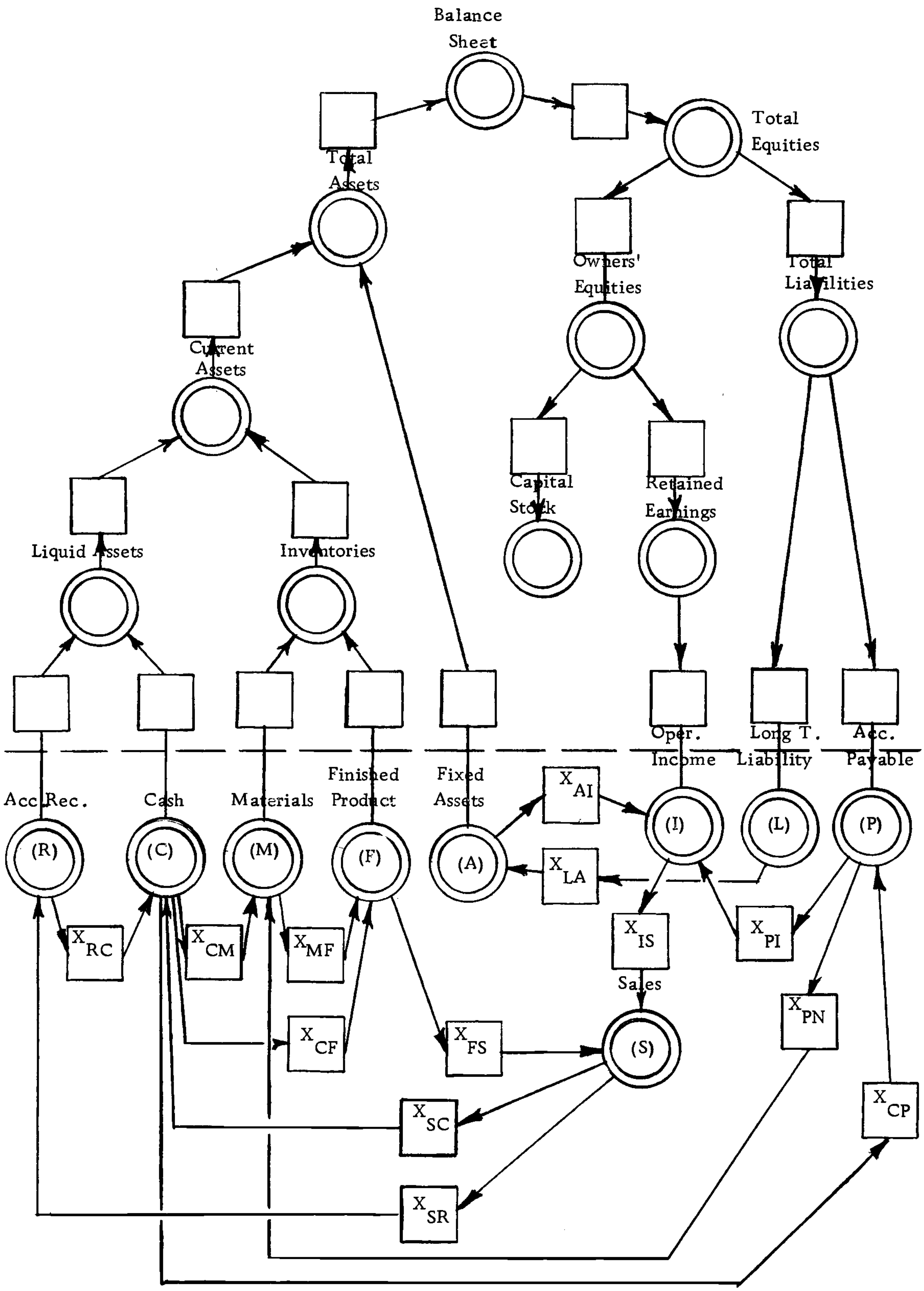


Figure 4.2. Integration of the IAF and HA network, or the HA-IAF network.

The left upper part corresponds to the asset side of the HA network, and the upper right to its equity side.

Figure 4.2 is not built completely on the IAF assumptions. It should be remembered that the HA network has its virtue in showing how the several levels of accounts are related by indicating how each lower level can be closed out to each higher level account. In other words, Figure 4.2 assumes an end-of-period time to trigger the HA flows; these closing transactions should be considered non-existent during the course of the period itself. Nevertheless, Figure 4.2 communicates the important consistency relationship needed between the two networks. It also helps in the construction of Figure 4.6, that integrates all the higher level accounts with the lower level ones used in an expanded IAF network. For each ST node of the IAF network section of Figure 4.2 it is possible to form a loop through the HA network to arrive back at the starting ST node. This loop is unique, and the arrow directions of the HA network are adapted to follow the loop direction indicated at the starting ST node of the IAF network. For example, the consumption of raw material for the production of finished products is shown in Figure 4.3. However, if a loop is forced to go through more than one account node of the IAF network in order to reach the HA network, it has to follow a directed path within the IAF network. This is the case for a loop starting at X_{FS} or X_{IS}. The reason is that the sales account is at a lower level than the operating income account, and for this reason it is always closed out to the latter one through transaction X_{IS}. The X_{FS} loop will leave the sales account going either through a cash or an accounts receivable transaction or a combination of the two. Figure 4.4 shows the two possible loops that can be followed with a sales transaction.

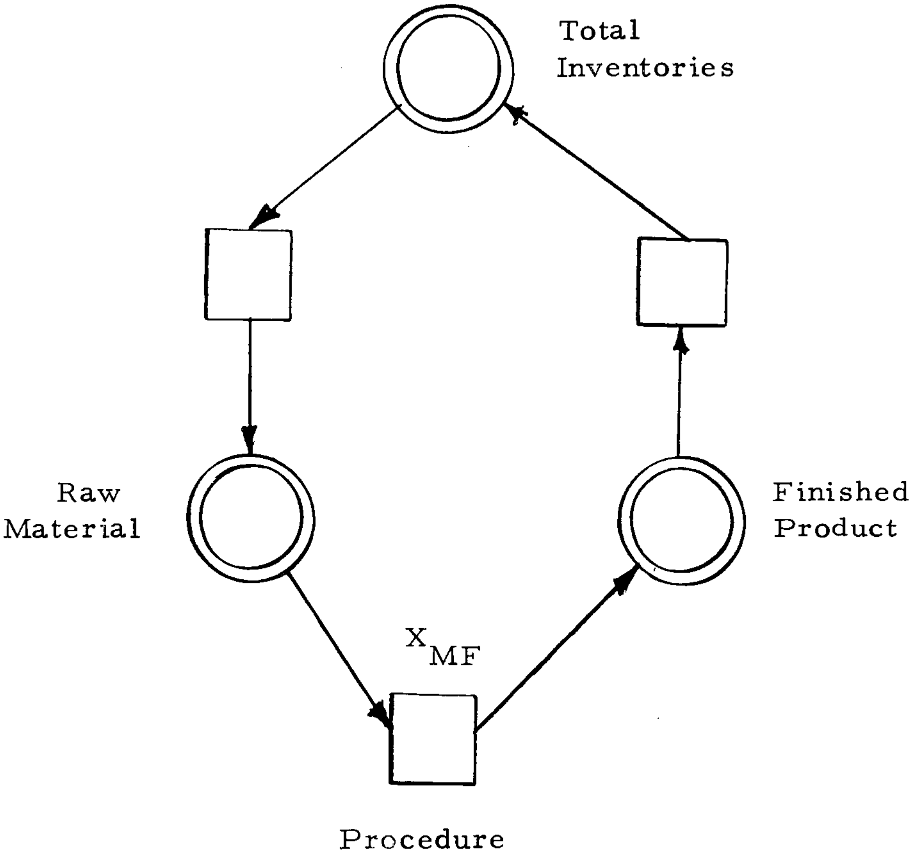


Figure 4.3 The X_{MF} loop.

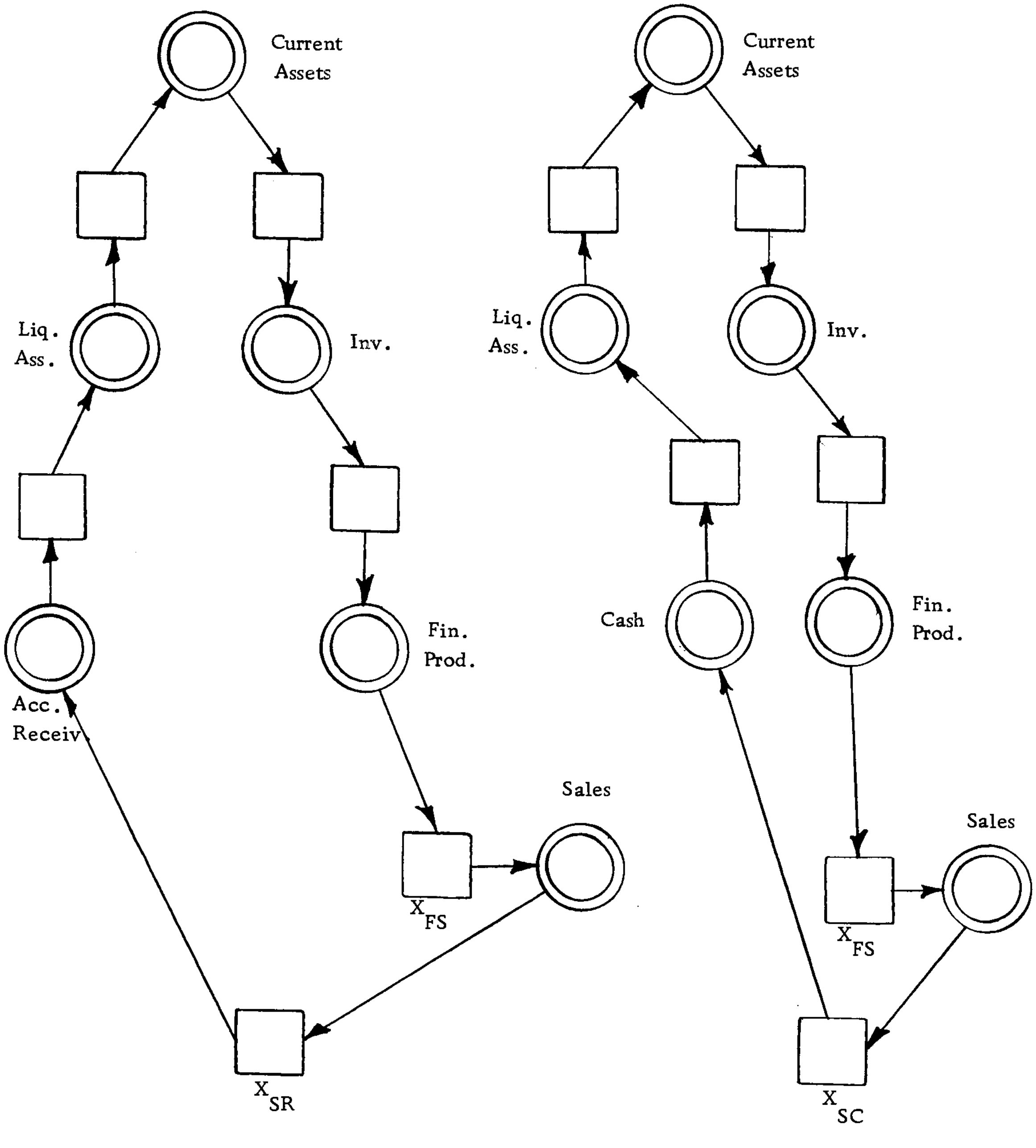


Figure 4.4. The X_{FS}-X_{SR} and the X_{FS}-X_{SC} loops.

Similarly, the contribution loop can go over the cash account or accounts receivables, depending on whether or not sales was in cash, on credit or a combination of the two. These two loops go over the asset and equity branches of the HA network. Only the cash X_{IS} loop is shown in Figure 4. 5; by changing the cash account for the accounts receivable and replacing the X_{SC} transaction by X_{SR}, the accounts receivable X_{IS} loop is obtained.

It is apparent that the simple transactions X_{SR} and X_{SC} can no longer be considered independent of the X_{FS} and X_{IS} transactions. Depending on their combinations, they may involve a different set of higher level accounts. In order to distinguish the four possible loops, we shall identify them as:

loop X_{FSSC} for the combination X_{FS} and X_{SC}.

loop X_{FSSR} for the combination X_{FS} and X_{SR}

loop X_{ISSC} for the combination X_{IS} and X_{SC}

loop X_{ISSR} for the combination X_{IS} and X_{SR}.

These four new loops may be considered as four new simple transactions, that eliminate the use of the sales account. This is very natural as the sales account is always to be closed out to the operations income account, that is one level higher.

Another method of dealing with the simultaneous updating problem in situations where at least one higher level account appears

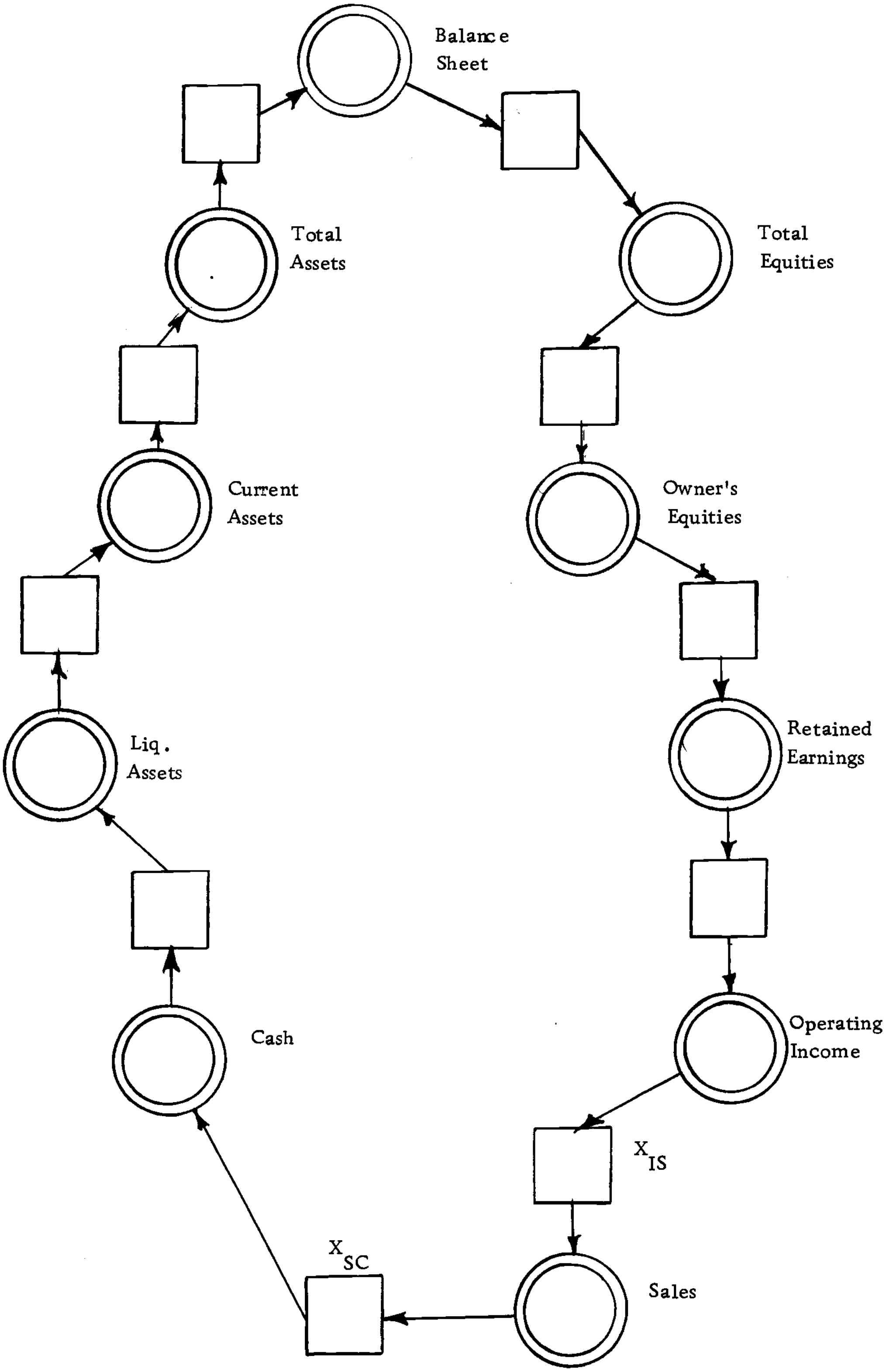


Figure 4.5. The Cash X_{IS} Loop.

in the IAF network, is to abandon the concept of loop formation, and instead update the higher level accounts of the involved transactions by closing them out until the root of the tree, the balance sheet, is reached. This means that the transaction X_{IS} is only debited for the sales account and credited for the income account, the retained earnings, the owner's equity account, and the total equities. The other transactions X_{FS}, X_{SR}, and X_{SC} do not involve accounts that are hierarchically dependent with respect to each other, so that all accounts along the debit path and credit path leading to the balance sheet node will have to be respectively debited and credited. It should be noted that this method is not as efficient as the method of loop forming since the total number of accounts to be updated increases from 15 accounts with the loop method to 31 using a pure closing out technique. However, the computation of X_{FSSC}, X_{SSR}, X_{ISSC}, and X_{ISSR} becomes much more complex either through use of the conversion matrix or the direct solution method of an over-determined model. Therefore, a combination of the loop method and the closing out method will be used for simultaneously updating all accounts; the loop method will be used for all transactions except X_{FS}, X_{IS}, X_{SC}, X_{SR}.

Using this information about loops, the integration between lower and higher level accounts is developed by drawing an expanded IAF network. The lower part of Figure 4.6 shows the previously

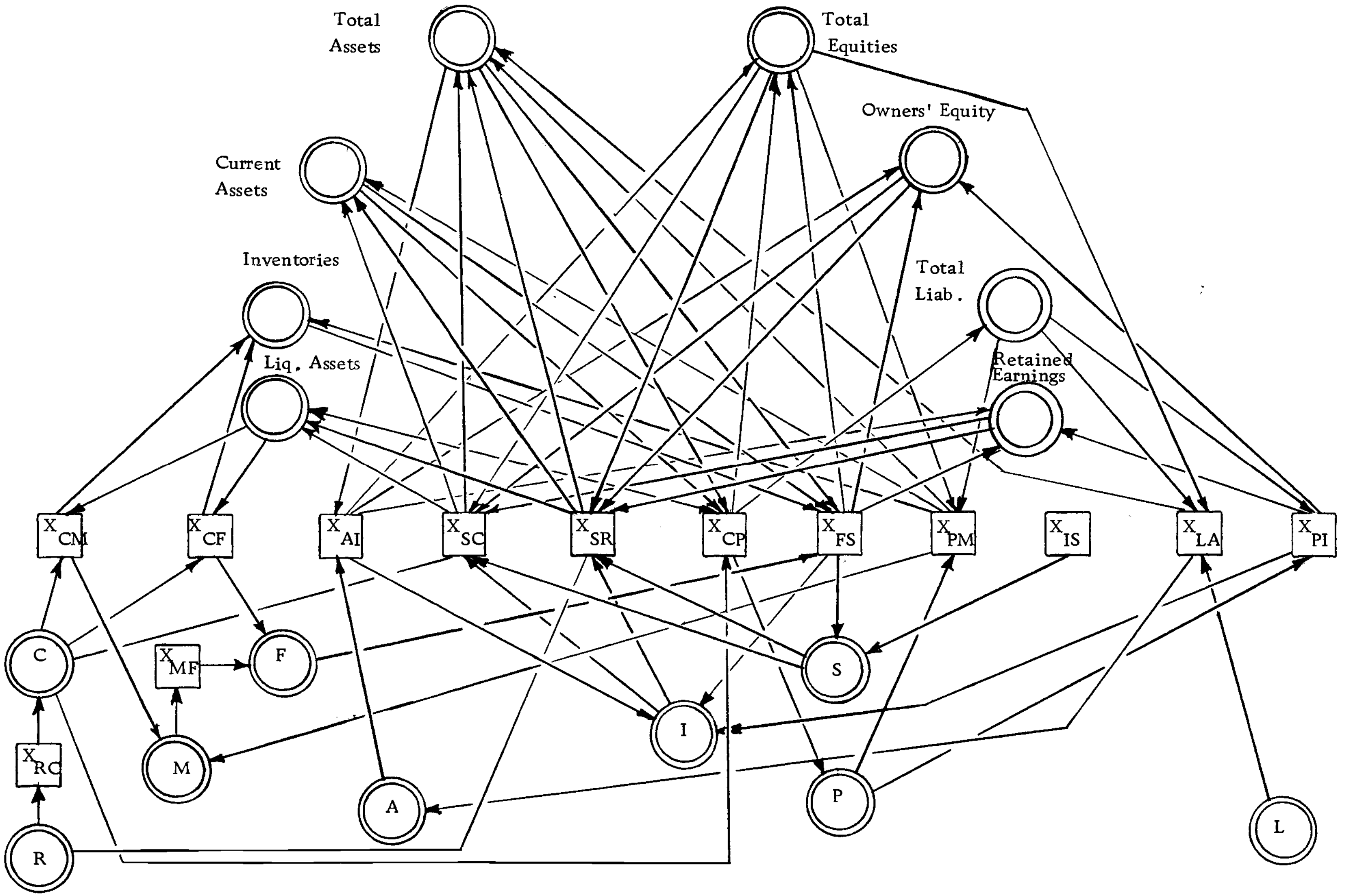


Figure 4.6. Expanded IAF network.

derived IAF network, while the part above the expansion includes all higher level accounts. The logical AND-relation of the activity nodes applies and means that all connected account nodes are simultaneously changed when the transaction node differs from zero. This network can be directly translated into an expanded incidence matrix, by using the conversion rules discussed earlier. Let $\hat{T}$ be the expanded incidence matrix as shown in Table 4. 3, and let column vector $\hat{B}$ indicate the value for each transaction (in the same order as the columns of the T matrix). The multiplication $\hat{T}*\hat{B}$ gives the total change over the period for each of the account balances.

$$\hat{B} = [\ 7,875 \quad 9,625 \quad 19,500 \quad 17,925 \quad 12,325 \quad 27,175 \quad 2,700 \quad 11,950 \quad 17,925 \quad 7,675 \quad 1,500 \quad 500 \quad 3,000]$$

$$\Delta\hat{Y} = \hat{T}*\hat{B} = [\ 2,000 \quad 0 \quad -2,000 \quad 0 \quad 1,500 \quad 0 \quad 2,200 \quad -.700 \quad -2,000 \quad 2,000 \quad 0 \quad 1,500 \quad -1,500 \quad -700 \quad 0 \quad 0 \quad -700 \quad 0 \quad 0 \quad -800 \quad -3,000]$$

Adding this change to the beginning balances results in the end-of-period balance for all lower and higher level accounts simultaneously.

$$Y_2 = Y_1 + \Delta Y = [\ 7,000 \quad 20,000 \quad 3,000 \quad 2,500 \quad 11,500 \quad 0 \quad -15,800 \quad -17,700 \quad 5,500 \quad 27,000 \quad 32,500 \quad 44,000 \quad -44,000 \quad -17,700 \quad 0 \quad 0 \quad -17,700 \quad 0 \quad 0 \quad 26,300 \quad -10,500]$$

Table 4.3. Expanded Incidence Matrix.

	X_{CM}	X_{PM}	X_{MF}	X_{RC}	X_{CP}	X_{FS}	X_{IS}	X_{SC}	X_{SR}	X_{CF}	X_{AI}	X_{PI}	X_{LA}
Cash	-1			1	-1			1		-1			
Acc. Receiv.				-1					1				
Inv. Materials	1	1	-1										
Inv. Fin. Prod.			1			-1				1			
Fixed Assets											-1		1
Sales						1	1	-1	-1				
Short T. Liab.		-1			1							-1	
Oper. Income						1		-1	-1		1	1	
Total Inv.	1	1				-1				1			
Liquid Assets	-1				-1			1	1	-1			
Current Assets		1			-1	-1		1	1				
Total Assets		1			-1	-1		1	1		-1		1
Total Equities		-1			1	1		-1	-1		1		-1
Owner's Equity						1		-1	-1		1	1	
Capital Stock													
New Investment													
Retained Earn.						1		-1	-1		1	1	
Taxes													
Dividends													
Total Liab.		-1			1							-1	-1
Long T. Liab.													-1

Just as we derived a LP model from the basic IAF network, it is possible to involve higher level account balances into the constraint set. For instance, if management should be interested in keeping the ratio of the current assets to the current liabilities between bounds, it would suffice to use the logical OR-relationship at the current asset and liability nodes to determine which ST nodes would be involved. By creating a resource node that connects all relevant ST nodes to the source such types of financial ratio constraints can be incorporated in the LP model.

V. DEVELOPMENT OF A DATA BASE AND THE RPMLPAS PROGRAM SYSTEM

Chapters II, III and IV studied the use of the RPM network technique as a communication tool between the information user and the model builder. The contemporary use of the computer in data processing and in the solution algorithms of (decision) models involves a third group of specialists referred to as the systems analyst-programmers. They are the people that develop the data base structure and program its applications. Their role is crucial in securing accessibility to existing data. In this Chapter we shall study the use of the RPM network technique as a communication tool between the model builder and the systems analyst-programmer.

More specifically, we shall first introduce some minor modifications to the standard RPM-LP network and set up some conversion rules leading to the determination of a data structure for LP algorisms. This illustration is general enough to suggest extension of its use to other management science techniques where RPM has proven useful (e.g., CPS/PERT, Branch and Bound techniques ...). Applied to the LP algorithm the data structure proposed has the advantage of providing a sparse matrix storage technique that is useful for the solution of large scale LP models.

The proposed derivation is not intended to be a cure all method, but intends to provide a basis to study the needed access paths to data elements. Indeed, the general procedure may provide data structures that are more complex than is required. In the second section, it will be illustrated that for the RPM-HA network (that is the tree type network) a data structure can be used that is simpler than the one needed for solving the LP model.

Finally, the general mode of operation of the RPMLPAS programming system that uses the data structures derived in sections one and two will be discussed. First, however, some concepts used in this chapter will be defined.

Most data base systems permit users to interact with the data in terms which are independent of the manner in which the data are physically stored. In such systems, the user's conception of the data is called a data structure (CODASYL, 1971). Data elements (fields) associated with each other, or likely to be processed together, are gathered into the same segment (IBM G320-1981, 1972). The same grouping criterion holds for collecting segments into records, and records into files. Therefore, the actual breakdown of a data structure is to some extent arbitrary.

Segments can be considered the "atom" part of a data base; a segment is the smallest unit, that can be accessed by a pointer, a segment possesses an identification name, and all segments with the same identification have the same size, not only within the record, but also within the entire data base. In case records can have a different number of repetitive (same) segments, records cannot be stored efficiently as a continuous string of data. The relationships between segments are therefore maintained by pointers, connecting the segments to one another, so that the physical disposition of segments is irrelevant.

To design a data structure, it is important that the possible paths of accessibility among different elementary "atoms" of data are specified (D'Imperio, 1969). The primal and the dual simplex method for solving LP problems require respectively access by column and by row. In each access, it is desirable for the multiplication or the inversion operation, to start with the first element of the column or row and to proceed the access to the next one, until the last element is encountered. On the other hand, in the accounting system, it is desirable to access all accounts of a loop each time a transaction occurs.

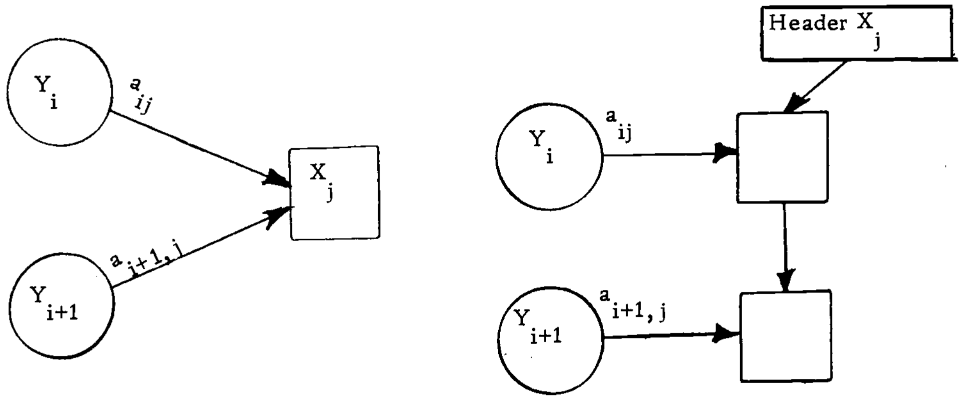
Data Structure for the RPM-LP Network

Three types of data structures will be introduced: a primal oriented data structure for the primal solution to the LP problem, the dual oriented data structure for the dual solution technique, and the primal-dual data structure for the combination of the primal and dual solution technique.

Primal Data Structure (or PDS)

The data structure proposed is based on a list structure approach for sparse matrices. The RPM-LP network has been defined (Chen, 1973) as a graphical presentation of input-output relationships among activities and resources within a system. A first conversion rule between an RPM network and its data structure is that there is a one to one relationship between a resource node and a resource record, and an activity node and an activity record. If an activity record is then considered to consist of a set of arrow segments linked together into a list structure, it becomes possible to derive a primal data structure for the LP model. In order to accomplish this structure, three cases of RPM connections are to be considered:

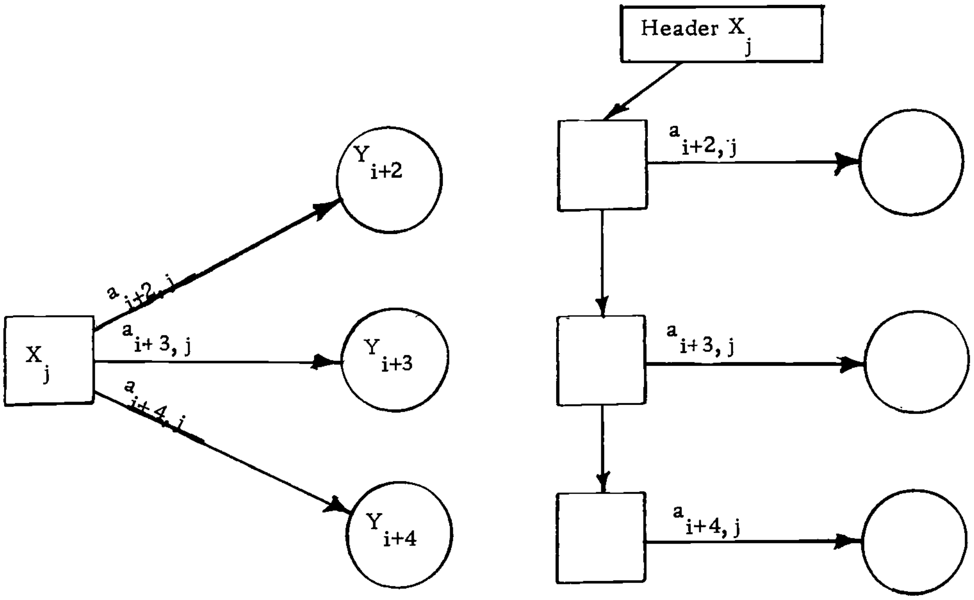
a. Several resource nodes connect an activity node from the left. Figure 5.1 shows the translation of the standard RPM into a list structure.



a. Standard RPM. b. List Structured RPM.

Figure 5. 1. Conversion of standard RPM into a list structured RPM for the from-the-left resource node structure.

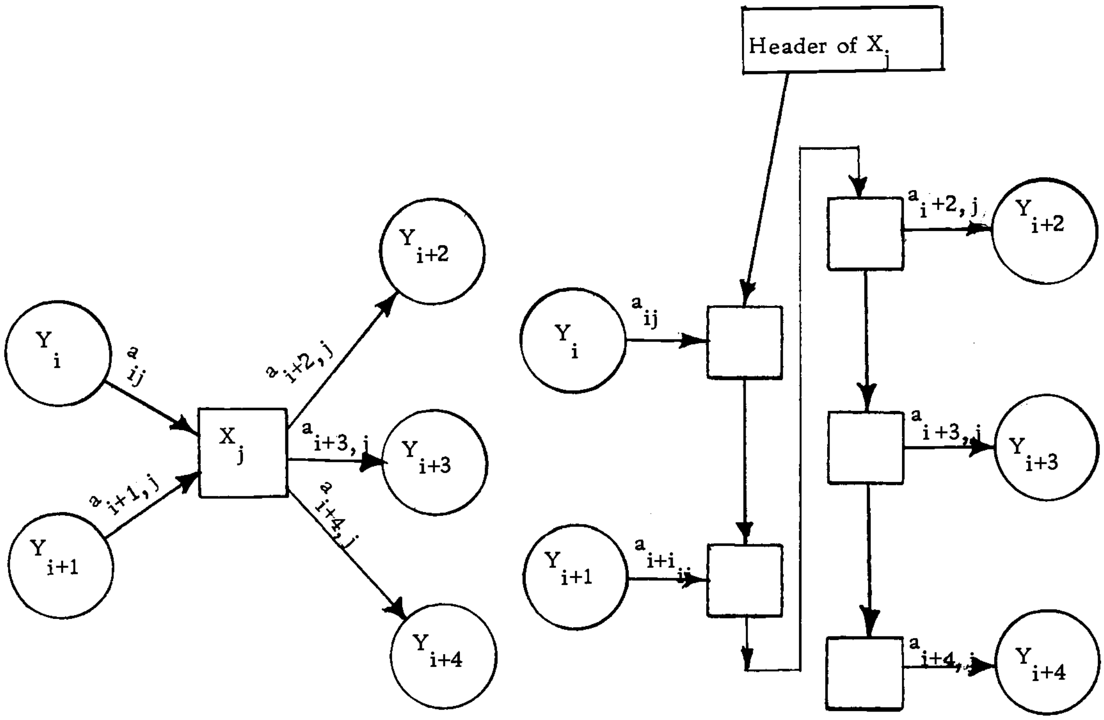
b. Several resource nodes are connected with an activity node from the right (see Figure 5.2).



a. Standard RPM. b. List Structured RPM.

Figure 5. 2. Conversion into list structured RPM for a from-the-right resource node structure.

c. Resource nodes are connected with activity node from both sides (right and left). This case is a combination of cases a. and b., as shown in Figure 5.3



a. Standard RPM. b. List Structured RPM.

Figure 5.3. Conversion of standard RPM structure into list structure for the general case (resource nodes to the right and the left of the activity node).

Figure 5.4 illustrates this conversion for the product mix example of Chapter II.

For the time being, this data structure will be limited to the information contained in a LP model. Because an arc exists only in a RPM-LP network when the coefficient (a_{ij}, b_i, or c_j) carried

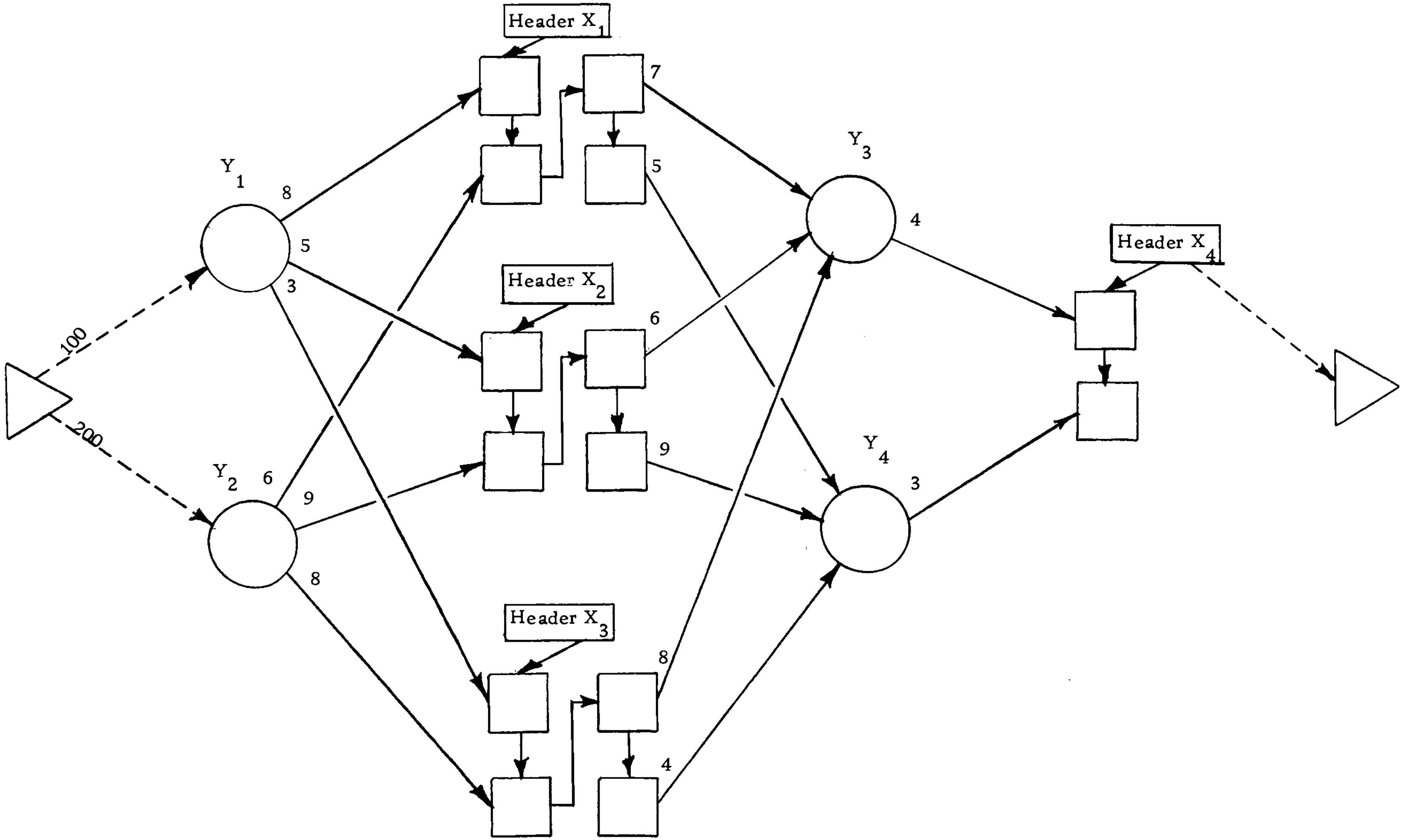


Figure 5.4. Primal list structured RPM-LP network for the product mix problem of example one.

by the arrow is nonzero, a close correspondence between a RPM-LP network and a packed form of the sparse matrices can be expected. If we neglect, for the moment, the storage of the inverse or the product form of the inverse of the basis, then an activity record has a structure as shown in Figure 5.5

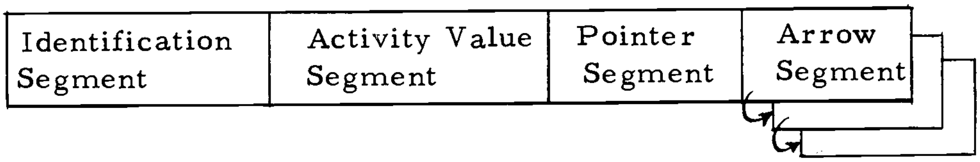


Figure 5.5. Layout for activity record.

The arrow segment is a repetitive segment, that corresponds with a linked square in the list structured RPM-LP model. Each non-zero data element a_{ij} corresponding to activity x_j needs only to be identified by its related resource node. A pointer in the third field points to the next arrow segment of the list, if the pointer value is nonzero. A zero value in the third field indicates the end of the list. The detail of an arrow segment is shown in Figure 5.6.

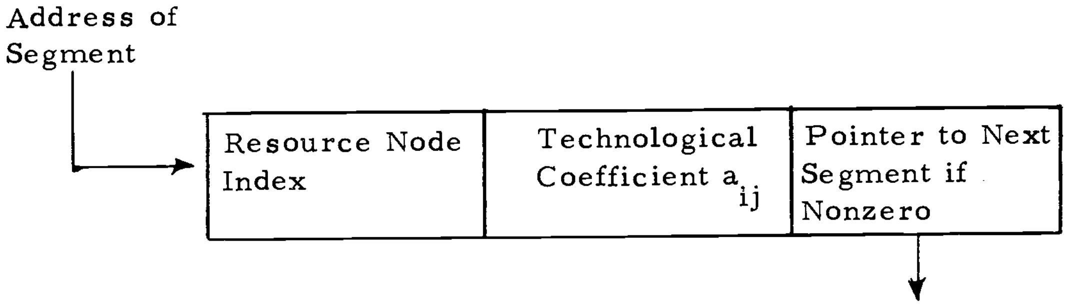


Figure 5.6. Layout for the arrow segment.

The pointer segment consists of one data element, pointing to the list of arrow segments associated with the activity node.

The remaining two segments (the identification and the activity value segment)are shown in Figure 5. 7.

Short Description of Activity

a. Identification Segment.

Cost Coefficient	Value of Activity	Opportunity Cost

b. Activity Value Segment.

Figure 5. 7. Layout of the fixed length part of the activity record.

The total storage for an activity record consists of two parts: the FPAR (Fixed Part of an Activity Record) and the SLL (Storage for Linked Lists). If the RPM-LP network has n activity nodes, the total FPAR storage will consist of n contiguous locations, each of which contains the first three segments of an activity record.

The principal advantage of the SLL storage scheme is that changes during the computations from non-zero to zero elements or from zero to nonzero elements can be easily accommodated without the need to change the physical storage of other segments in order to preserve the correct data order. This feature will be useful in order to store the inverse or product form of the inverse in packed matrix form. Assuming that for each column of the inverse one record exists, then one segment per non-zero column element (similar to the arrow segment) can be used (see Figure 5. 8). All segments of one record can be linked into one linked list that can be stored into the SLL, provided we have a header outside the SLL pointing to the beginning of the inverse matrix.

Row Index i	Value of Matrix Element	Pointer to Next Segment

Figure 5. 8. Layout of a segment for a column record from the inverse matrix.

Another prime advantage of the list structure is the possibility of changing the lengths of records. This feature is useful for postoptimality studies, in case an analyst is interested in adding one or more activity nodes or constraints to the model.

The available storage for linked list segments can be maintained as a chained list, where the third field in each available segment points to the next one to form the available list. The last segment of storage can contain a zero value in the third field. Also for this list, the address of the first available segment will have to be stored outside SLL. When a segment is needed, it can be "popped off" the top of the available list, or a segment can be added to the top of the available storage each time a non-zero element of the SLL becomes zero.

Resource Record

For the primal data structure of a RPM-LP network, the resource record shows a rather simple structure as shown in Figure 5.9.

Identification Segment	Resource Value Segment	Pointer Segment

Figure 5.9. Layout of the resource record in a Primal Data Structure of RPM-LP network.

The identification segment contains the name of the constraint (Y_i), while the resource value segment is shown in Figure 5.10.

Initial Value of a Resource	Slack Value of a Resource	Shadow Price of a Resource

Figure 5.10. Layout of the resource value segment for a Primal Data Structure of RPM-LP network.

The pointer segment contains a pointer to the corresponding column of the inverse of the basis. In case the product form of the inverse is used, a zero value in the pointer field would refer to a regular column of the identity matrix, while a nonzero value would refer to the unique column of the identity matrix.

Table 5.1 is a representation of a data structure for the primal list structured RPM-LP network for example one.

Dual Data Structure (or DDS)

The Dual Data Structure (hereafter DDS) is based on a modification analogous to that of the primal of the RPM-LP network. Instead of being centered around the activity node, a list structure is formed at the resource node. The DDS will allow computations to be performed row-wise, as required by the dual simplex solution technique. Figure 5.12 shows the dual list structured RPM-LP network.

The principles used to derive the DDS from a list structured RPM-LP network are almost the same as for the PDS. The main difference is that the two last segments of an activity record from the PDS will be added to the resource record of a DDS. The two records of the DDS are shown in Figure 5.11.

Identification Segment	Activity Value Segment

a. Activity Record.

Identification Segment	Resource Value Segment	Pointer Segment	Arrow Segment

b. Resource Record.

Figure 5.11. Segments of the activity and resource record for a Dual Data Structure.

Table 5.1 Primal Data Structure of initial data for example one.

Fixed Part of Activity Record.

Identification Segment	Activity Value Segment			
Variable Name	Cost Coefficient	Value of Variable	Opportunity Cost	Pointer Segment
X_1	0	0	0	100
X_2	0	0	0	102
X_3	0	0	0	104
X_4	1	0	0	106

Fixed Part of Resource Record.

Identification Segment	Resaurse Value Segment			Pointer Segment
Resource Name	Initial Value of Resource	Slack Value	Shadow Price	Pointer to Inventory
Y_1	100	100	0	114
Y_2	200	200	0	115
Y_3	0	0	0	116
Y_4	0	0	0	117

Storage for Linked Lists.

	Pointer Res. Node	Tech. Coeff. or Elem. of Inv.	Pointer to Next Segm.		Pointer Res. Node	Tech. Coeff. or Elem. of Inv.	Pointer to Next Segm.
100	1	8	107	113	4	3	0
101	3	-7	108	114	0	0	0
102	1	5	109	115	0	0	0
103	3	-6	110	116	0	0	0
104	1	3	111	117	0	0	0
105	3	-8	112	118	0	0	0
106	3	4	113	119	0	0	0
107	2	6	101	120			
108	4	-5	0	121			
109	2	9	103	122			
110	4	-9	0	123			
111	2	8	0	124			
112	4	-4	0	125			

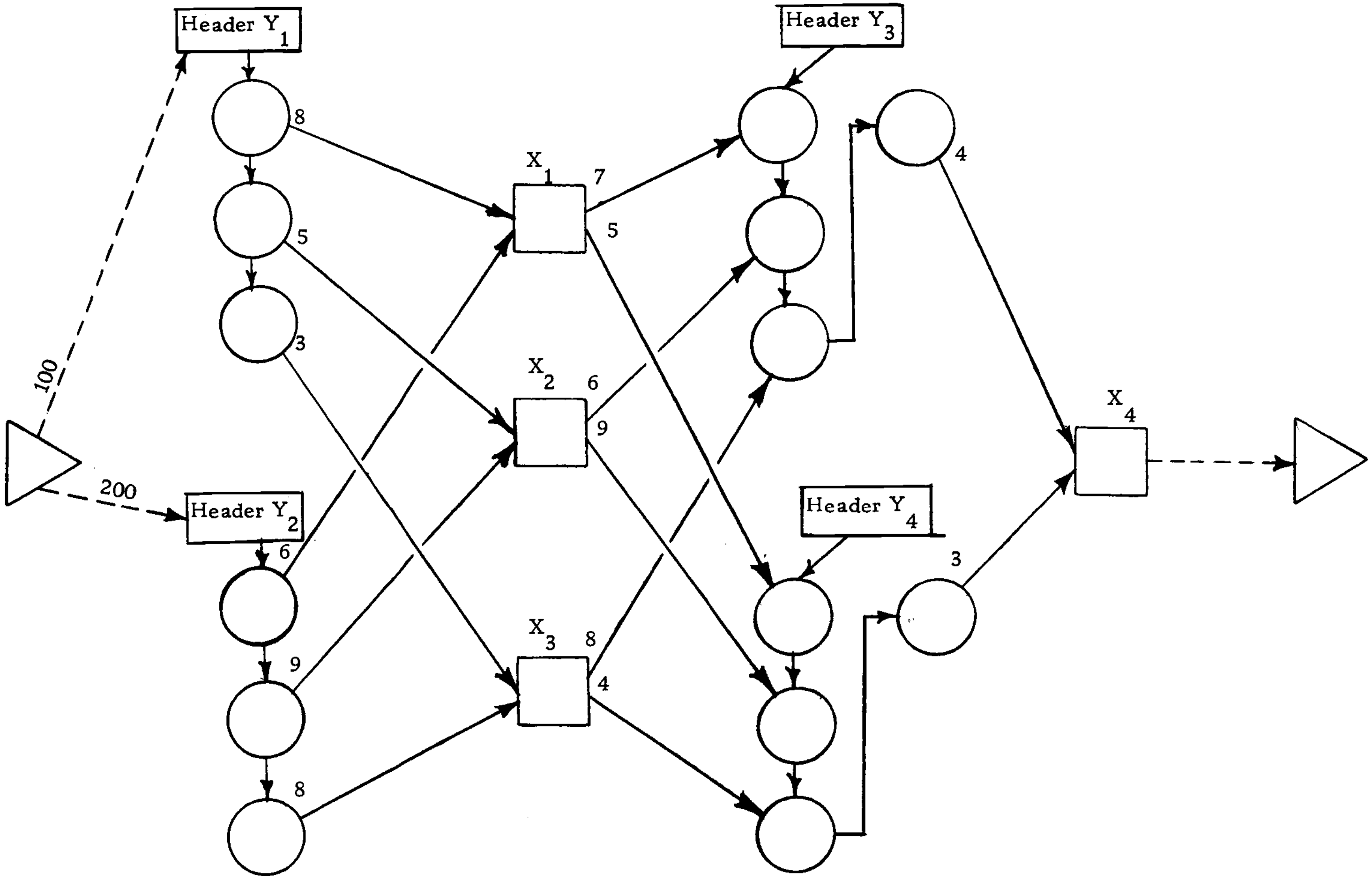


Figure 5.12. Dual List Structured RPM-MS.

Table 5.2. Dual Data Structure of initial data for example one.

Fixed Part of the Activity Record.

Identification Segment	Activity Value Segment		
Variable Name	Cost Coefficient	Value of Variable	Opportunity Cost
X_1	0	0	0
X_2	0	0	0
X_3	0	0	0
X_4	1	0	0

Fixed Part of the Resource Record.

Identification Segment	Resource Value Segment			Pointer Segment	
Resource Name	Initial Value	Slack Value	Shadow Price	To Inv.	
Y_1	100	0	0	114	100
Y_2	200	0	0	115	102
Y_3	0	0	0	116	104
Y_4	0	0	0	117	105

Storage for Linked Lists.

	Linked List Segments				Linked List Segments		
	Pointer to Activity Node	Techn. Coeff. Value Inv.	Pointer to Next Segm.		Pointer to Activity Node	Techn. Coeff. Value Inv.	Pointer to Next Segm.
100	X_1	8	106	111	X_3	-8	102
101	X_1	6	108	112	X_2	-9	113
102	X_4	4	0	113	X_3	-3	103
103	X_4	3	0	114	0	0	0
104	X_1	-7	110	115	0	0	0
105	X_1	-5	112	116	0	0	0
106	X_2	5	107	117	0	0	0
107	X_3	3	0				
108	X_2	9	109				
109	X_3	8	0				
110	X_2	-6	111				

The arrow segment in the DDS differs from the PDS by the first field, that now contains an index of the activity record. Table 5.2 gives an illustration of the dual data structure for the product mix problem.

Primal-dual Data Structure (or PDDS)

The Primal-dual Data Structure (hereafter PDDS) is a combination of the PDS and the DDS. By this data structure the original problem data, usually summarized in the simplex tableau, can now be accessed in column and in row order. Figure 5.13 shows how the list structures are built around both the resource and the activity node for the product mix example.

It is possible to derive a data structure for the PDDS in a similar manner to that for the PDS and the DDS. Such a data structure can be formed by combining Tables 5.1 and 5.2; this implies the duplication of all technological coefficients and the use of n+m+1 lists. Despite the fact that all technological coefficients are duplicated once, this data structure can be considered acceptable because the original model data has to be retained, so that the possible additions or deletions of activity or resource nodes should not be too complex a process.

It is also possible to eliminate this duplication in technological coefficients by creating a multi-list structure. Only one arrow

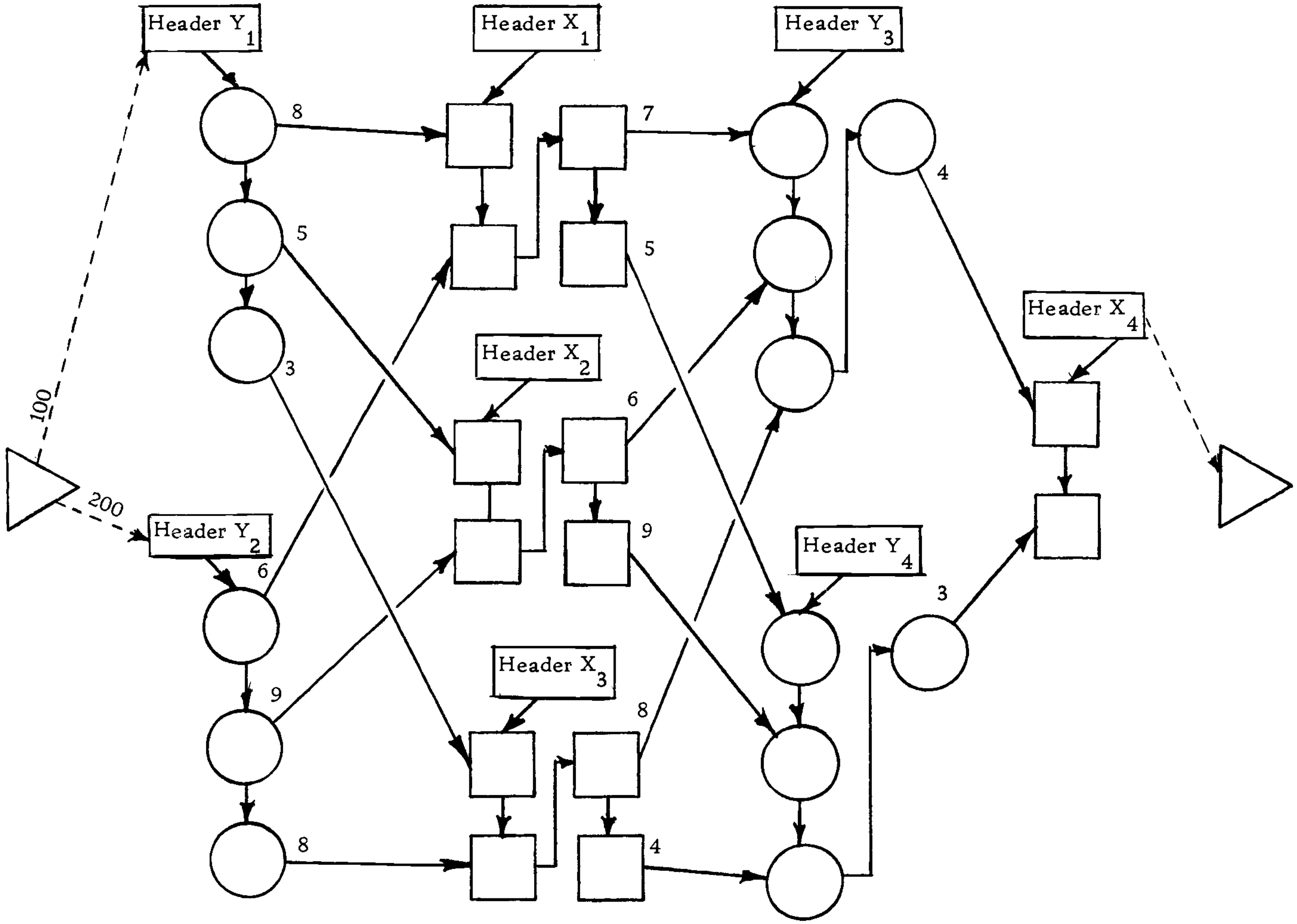


Figure 5.13. Primal-dual Data Structure for the RPM-LP network.

segment, that is shared by the activity and resource record is created for each technological coefficient. Technically speaking, the arrow segment becomes an arrow record that is referenced by the activity and resource record. Figure 5.14 shows the layout of the arrow record.

Resource Node Index	Activity Node Index	Technological Coefficient	Primal List Pointer	Dual List Pointer

Figure 5.14. Layout of the arrow record.

Construction, deletion and insertion of segments for a double list is not such a simple issue as for a single list structure. For example deleting a segment of a single list is not difficult, since the segment is initially accessed from the preceding segment. However, if a segment or record participates in a double list, such as needed for the PDDS of Figure 5.17, the predecessor of the list, that is not accessing the matrix element is not known. This problem can be solved by tracing the predecessor from the beginning of the second list, or more efficiently, by using extra pointers to keep track of the predecessor address.

This last data structure has been applied in the programming of the RPMLP subroutine (to be discussed in the third section of this chapter). Appendix III illustrates the linked list as applied by this subroutine to example one.

Data Structure for the RPM-HA Network

In principal, the hierarchical accounting tree proposed in Figure 2. 1 can be considered as a network, and a methodology (similar to that for the RPM-LP network) of creating a data structure can be followed. Account records could correspond to the circles and closing transaction records to the squares. Similarly, a double list structure for the arrow segment would allow access to successive account balances and closing transactions.

However, two differences exist with respect to the RPM-LP network. First, in a tree network a lower level node can never be connected to more than one higher level node, while the coefficient on the arrows always equals one. Second, it is desirable to access only a partial or complete loop of accounts in the RPM-HA network. A complete loop of accounts can be constructed by departing from an activity node in the RPM-LP network in the debit and credit direction of that node and climbing through the branches of the tree until a loop is formed. On the other hand, a partial loop is formed by going from an activity node in the RPM-LP network to the balance sheet node. These features eliminate the need for closing transaction records and arrow segments provided that the tree structure is built in the account node record. Figure 5. 15 shows the account record used in the RPMLPAS programming system (see next section).

Account Identification	Account Balance	Pointer to Next Higher Level Acc.

Figure 5.15. Layout of the account record for the RPM-HA network.

For each higher and lower level account exists such a record, except for the balance sheet node for which no record is needed. This simplified data structure not only uses less memory space, but allows faster account updating than would be the case with a data structure involving account, closing transaction and arrow records.

Appendix IV explains how the hierarchical network is entered before it is converted into the data structure of Figure 5.15.

The RPMLPAS Programming System

In this section we shall describe the programming system RPMLPAS and illustrate the implementation of example two. This includes the solution of the model as it has been built through the analysis of Chapter III, the transformation of the LP solution into accounting transactions as studied in Chapter IV, and the updating of account balances of the accounting system as described in Chapters II and IV.

This programming system is written in FORTRAN IV and implements the data structure derived from the RPM-LP and the

RPM-HA networks as described in the previous sections of this Chapter. Only one-dimensional arrays are used to allow fast addressing of data elements. The main program with its 17 subroutines occupies 3,971 words and uses a common data areas of 8,166 words. The program was implemented on the OS-3 operating system (on a CDC-3300). More detailed information needed for the actual implementation of the program is contained in Appendix IV.

Figure 5.16 shows a flowchart of the overall operation of the RPMLPAS system. At the start of the execution, the main program reads in three separate control cards: one for the LP model, one for the interface, and one for the accounting system. A read subroutine (SUBROUTINE PACK) is used to read in the LP model data and meanwhile packs this data according to the primal dual data structure of Chapter V. A second read subroutine (SUBROUTINE INTACC) reads in the interface and accounting data. After these read operations, the initial LP model data is transferred (in packed form) onto tape by subroutine STORE, and then the LP model is solved by subroutine RPMLP. If a solution counts subroutine CONVERT buffers the initial LP data back into the LP data storage area, and addresses the needed coefficients for converting the LP solution to transaction amounts. Finally, subroutine UPDATE is used to update the beginning-of-period account balances, after which

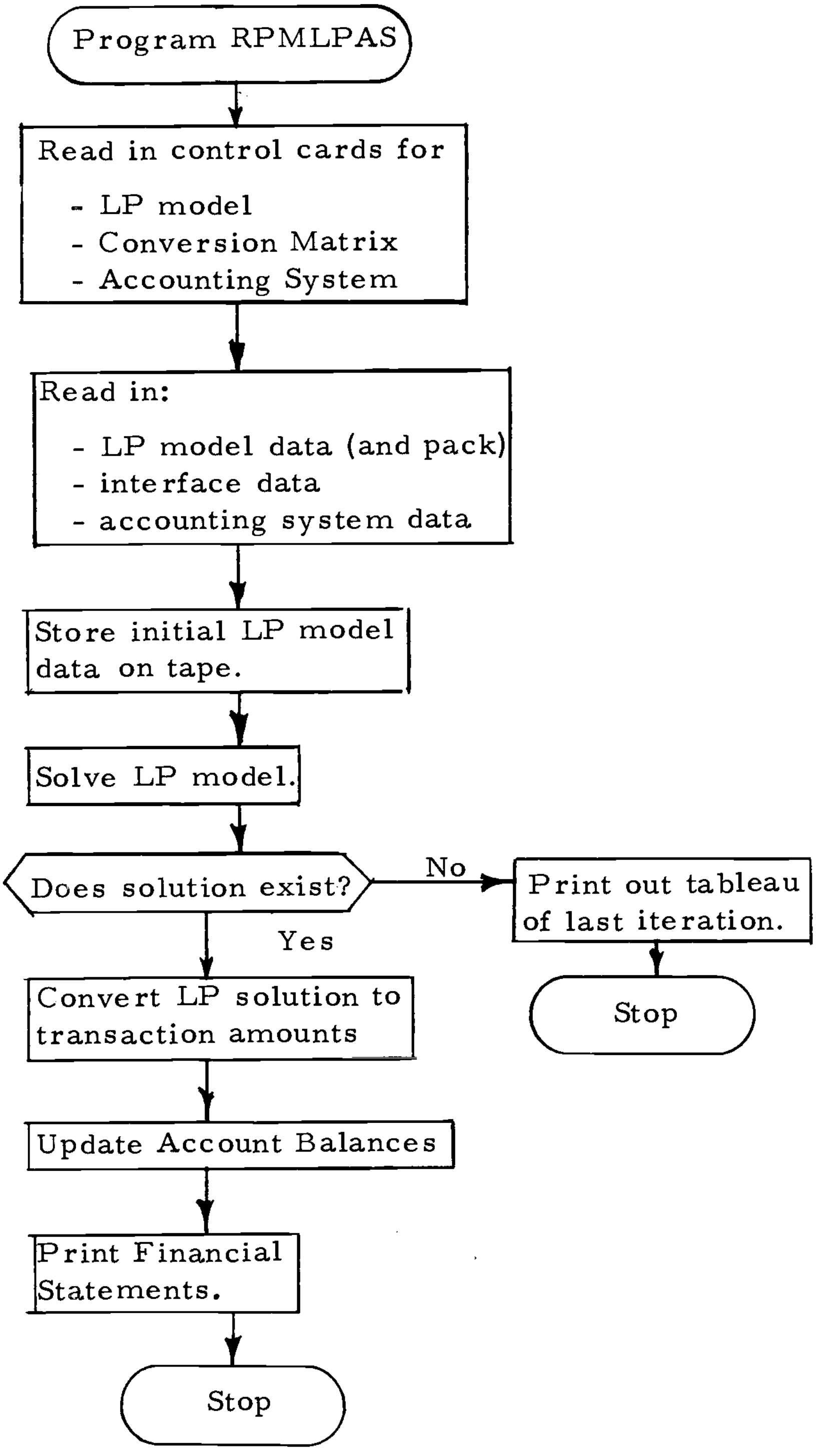


Figure 5.16. Flowchart of the general mode of operation for the RPM-LP-AS system.

subroutine FINSTAT prints out the financial statements as shown in Table 5. 4.

The algorithm used to solve the LP problem is a specialized algorithm based on the dual simplex method. It was suggested by Llewellyn (1964), who states that he has found the algorithm to be more efficient than any other he used, and given the name MINIT (MINimum ITerations). The algorithm distinguishes two phases. In Phase I, all equality constraints are solved first, as they belong to the set of defining equations. This feature of the algorithm considerably reduces the number of iterations involved since no artificial variables are used to determine a starting solution. In Phase II, both the primal and dual iterations are made. Both Phase I and Phase II choose the pivot that maximizes the rate of increase (decrease) in the objective function during a primal (dual) iteration. The details of the algorithm and a discussion of the theoretical reasons for its computational efficiency are described by Llewellyn (1964). Subroutine RPMLP is based on this algorithm and applies the special features for of a linked list data structure proposed in this Chapter[7].

---

[7]Salazar and Sen published an ALGOL code for the MINIT algorithm in the CACM (1968, Algorithm 333). This code and later corrections in CACM (Messham, 1969; Kolm and Dahltrand, 1971; Obradovic, 1973; Holmgren, Obradovic and Kolm, 1973) were referenced to program RPMLP in FORTRAN IV.

After the model has been optimized, it is desirable to access the initial problem data needed in the conversion process of LP results into transaction amounts. This initial data can be stored on tape or on disc before starting the optimization algorithm. RPMLPAS stores the initial data on tape, as its storage capacity is much bigger. The possibility for expanding the RPMLPAS system for multiple runs was the main reason for deciding on the use of tape media.

One of the main characteristics of an integrated information system is the elimination of data duplication. Because the conversion matrix multiplies the RPM-LP results with coefficients that appear in model data, its data structure has been reduced to a reference system. One record, as shown in Figure 5.17 is used for each nonnegative element of the conversion matrix. In case the coefficient of the matrix should be different in sign from the corresponding model coefficient, a minus sign is to be put before the row index of Figure 5.17. On the other hand, when an identity relationship exists between the value of an RPM activity value and a transaction amount, a zero value for the row index is entered.

Transaction Identification	Resource Number of Coefficient	Variable Number of Coefficient

Figure 5.17. Data structure of conversion matrix record.

The results of the model solution, and the derived transaction amounts have been summarized in Table 5.3. Table 5.4 shows a computer listing of the financial statements by the RPMLPAS program.

Table 5.3. Summary of the LP model results and the budgeted transactions.

LP Solution	LP derived Transactions	Remaining Transactions
X_{RC} 17,925	X_{CM} = 7,875	X_{AI} = 1,500
X_{CP} 12,325	X_{PM} = 9,625	X_{PI} = 500
X_{SCR} = 29,875	X_{MF} = 19,500	X_{LA} = 3,000
X_{CF} = 7,675	X_{RC} = 17,925	
X_{CPM} = 17,500	X_{CP} = 12,325	
X_{PI} = 2,700	X_{FS} = 27,175	
x_1 = 6,250	X_{IS} = 2,700	
X_2 = 3,500	X_{SC} = 11,950	
X_{IS} = 2,700	X_{SR} = 17,925	
X_{FS} = 27,175	X_{CF} = 7,675	

Table 4.5. Financial statements.

BALANCE SHEET

	BEGINNING BALANCE	ENDING BALANCE
ASSETS		
CURRENT ASSETS		
LIQUID ASSETS		
CASH	5000.00	7000.00
ACCOUNTS RECEIVABLE	20000.00	20000.00
TOTAL LIQUID ASSETS	25000.00	27000.00
INVENTORIES		
RAW MATERIALS	5000.00	3000.00
FINISHED PRODUCT	2500.00	2500.00
TOTAL INVENTORIES	7500.00	5500.00
TOTAL CURRENT ASSETS	32500.00	32500.00
FIXED ASETS (LESS DEPRECIATION)	10000.00	11500.00
	----------	----------
TOTAL ASSETS	42500.00	44000.00
	==========	==========

	BEGINNING BALANCE	ENDING BALANCE
EQUITIES		
LIABILITIES		
ACCOUNTS PAYABLE	18000.00	15800.00
LONG TERM LIABILITIES	7500.00	10500.00
TOTAL LIABILITIES	25500.00	26300.00
EQUITY	17000.00	17700.00
	----------	----------
TOTAL EQUITIES	42500.00	44000.00
	==========	==========

BUDGETED INCOME STATEMENT

SALES	29875.00
COST FINISHED PRODUCTS SOLD	27175.00

GROSS PROFIT	2700.00
FIXED COST	500.00
DEPRECIATION	1500.00

NET INCOME	700.00
	==========

BUDGETED STATEMENT OF FUNDS FLOW

INCOME FROM OPERATIONS		
NET INCOME	700.00	
DEPRECIATION	1500.00	

GROSS INCOME		2200.00
INCOME FROM OTHER SOURCES		3000.00
TOTAL SOURCES		5200.00
APPLICATION OF FUNDS		3000.00

INCREASE IN WORKING CAPITAL		2200.00
		==========

VI. CONCLUSIONS AND RECOMMENDATIONS

In this dissertation, we have represented the accounting information system and the linear programming decision model by RPM networks that clearly show their interdependence. It has been demonstrated that the RPM network technique is a potential communication device between the systems analyst-programmer and the management scientist by the provision of an RPM-derived data structure. These two issues have logically led to the development of the RPMLPAS programming system, that accepts an hierarchical accounting structure, beginning balances for accounts, an LP decision model, and transactions based on decisions beyond the LP decision domain, in order to provide financial statements that report the accounting status corresponding to the optimal LP solution, within the framework of the LP decision model. This approach allows a meaningful interpretation of model results by management. As pointed out throughout this dissertation, the construction of interdependent decision model and information system, requires a level of abstraction from the real organization, with its decision and responsibility structure. Further research will help in alleviating the problem of relating model decisions with actual improvement. A first step would be the implementation of a control function on the LP decision model.

The control function is inextricably tied to the planning function. We must provide feedback, so that outcomes are evaluated and action toward improvement can be taken. The purpose of the information system should be to create consistency between the process planning and process control. This means that the same set of procedures and assumptions should be used in developing a feedback information system as the procedures and assumptions used in the planning process. The interdependence between the RPM-HA and the RPM-LP networks, which we developed, can be of value for identifying the critical information elements.

A next step is to hold management accountable by generating a report of responsibilities for the deviations. In case of an hierarchical organization, it may be possible to relate the organization to the LP model in much the same way as the hierarchy of accounts has been related to it. The resource nodes of our RPM network can be termed as management nodes, while the activity nodes refer to the assignment of variances[8] from the LP model results to the several management nodes. In case of a matrix organization, these deviations will have to be aggregated in the several matrix dimensions.

---

[8]By variance, we refer to deviations between budgeted values and actual values. This includes monetary as well as non-monetary deviations.

The principle seems simple; however, managers should not be held responsible for uncontrollable variables. For example, demand will usually be an uncontrollable variable for the production manager, but partially controllable variable for the marketing manager who decides on the salesforce, advertisement allocation and the price setting. In general, decision variables of a LP model are controllable, while the b_i parameters are usually uncontrollable, while the a_{ij} parameters describe predicted relationships (or standards) among controllable and uncontrollable variables.

Not all of the differences between actual results and an optimal plan are assignable to the line roles of production, sales and financial executives. Errors caused by service departments should be separated before attempts are made to divide deviations from the optimal plan among the several line responsibilities. Most integrated models only represent the production, sales and related financial activities because they are the most essential and fortunately usually the best structured. It is useful to look at Figure 1.2 from the viewpoint of different departments that are involved in the "model solution and updating loop", and to identify the type of errors they may cause. See Figure 6.1 for this identification.

In summary, six types of errors should be identified. Forecasting errors which exist when parameters fall outside their

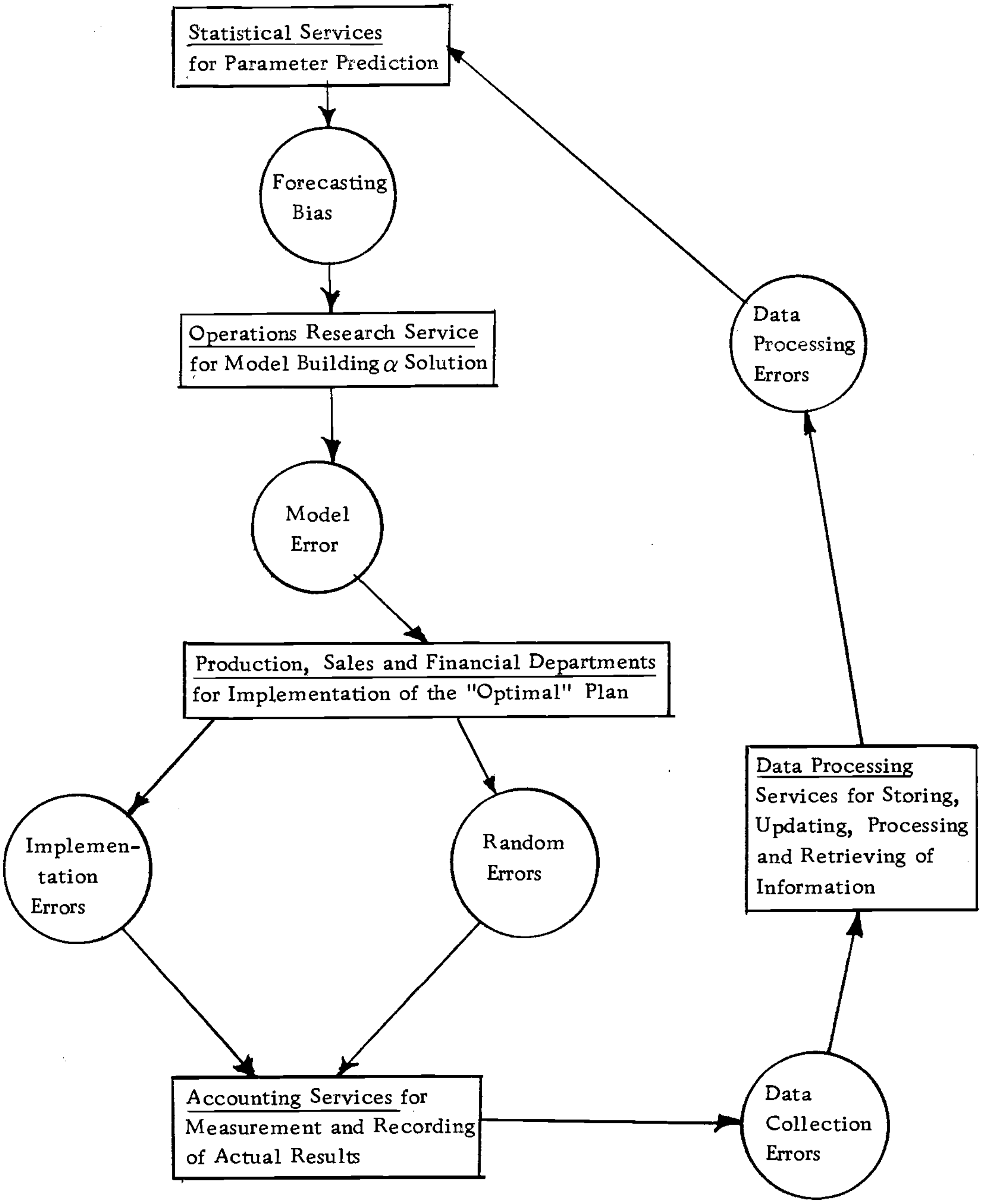


Figure 6.1. Error identification in model solution and updating loop.

confidence limits. Model errors referring to a structural error in the model (e.g., missing constraints or missing activities). Random errors are uncontrollable errors due to the course of nature in actual outcomes of activities. Implementation errors, that are performance errors due to inefficiencies at the operating levels and are accountable to line executives. Data collection errors caused by mistakes in the fundamental measurement process and/or in the recording of these measurements. Finally, data processing errors that can be caused by storage and retrieval mistakes, logic errors in programming, and rounding-off errors in computations with fundamental measurements.

Important simplifications with respect to forecasting have been introduced in the discussion of Figure 6.1; these are not always appropriate. Forecasting of parameters (e.g., expected sales or demand for a product) is frequently accomplished by inquiries to people with line responsibilities (e.g., sales manager and field salesmen). Evaluation systems have been formulated (Ijiri, Kihard, Putney, 1968), so that persons who are preparing forecasts are also held responsible for operating according to their forecasts. An incentive plan was proposed by these authors in which maximum reward was given for correct forecasts and penalty coefficients were used for over and under estimation deviations with respect to the actual results. Obviously, forecasting of the model parameters can be accomplished by a combination of techniques, and the evaluation system being introduced will achieve more accurate forecasting. However, this does not change the basic path of the model solution and updating process, and the deviations that may originate should be identified.

Another desirable area of research for the control function is the estimation of confidence limits for the activities and objective function of the model, based on the confidence limits of all parameters of the model. This could lead to improved reporting on the selected optimal plan and to better detection of forecasting and model errors. Control charts, similar to the ones employed in quality control, can be used for the identification and estimation of random errors. Models have been proposed to balance the cost of investigation and action triggering versus the present and future benefits, using the present value method (Duvall, 1967; Dyckman, 1969; Ozan and Dyckman, 1971). However, such models cause an information problem themselves, and are therefore not yet operational.

Thus, with respect to this function of reporting and performance analysis, a report generator for the RPMLPAS program can be built that will analyze a report on several types of errors by comparing the optimal solution of a model with data of actual events. Furthermore, the implementation error can be allocated to the several responsibility centers in consistency with the model structure.

One of the serious restrictions in the planning function is the single period model proposed. Two ways exist for planning in a multi-period dimension. First, the one-period model can be repeated over successive periods by a repeated solution of the proposed model using forecasted parameters for each of these periods. Despite the advantage that this method can be readily implemented on the RPMLPAS system, it has the disadvantage of ignoring the optimization that accounts for interperiod constraints. Important decisions, such as the level of inventory to be carried over from period to period which depends on the relative costs of producing and holding these products in successive periods are ignored. The optimization function would be better accomplished by a multi-period model that explicitly relates the activities of alternative time periods. Our one-period model can then be considered as a sub-problem of this multiperiod model, while interperiod constraints account for the longer term decisions. The RPMLPAS program of Appendix IV has the advantage of becoming relatively more efficient, the sparser the LP input matrix gets, and thus potentially useful for multiperiod models. However, further research in the use of the RPM network techniques for presenting interperiod constraints and their interfacing with the single period subproblem is desirable.

In this dissertation, we were not directly concerned with the measurement function of objects and events. It seems relevant to remark that the dual variables of an LP solution automatically determine the shadow price of constrained resource in accordance with the opportunities refected in the LP model. Research in this area may lead to an improved method of asset valuation and possibly to an "opportunity report", explaining bottlenecks and reporting the impact of changes in variables and coefficients with respect to the objective function and the financial statements. It should be noted that the RPMLP subroutine of the RPMLPAS programming system only computes the dual variables corresponding to the inequality constraints since no artificial variables are used for equality constraints. Fortunately, the dual solution vector for LP models with equality constraints can always be computed by the multiplication of $C_B B^{-1}$, where C_B is the vector of objective function coefficients of variables in the basis and B^{-1} is the inverse of the basis.

With respect to the RPMLPAS programming system in its present form, some practical restrictions still remain. First, the programming system does not provide the scaling feature desirable for the solution of LP models, so that coefficients with a large difference in absolute value will cause less rounding errors. Scaling of coefficients is assumed to be done by the user before the data is being entered into the system.

Second, the implementation of parametric programming is a desirable feature in order to make the system practical and useful for the planning and control function of decision making. Preferably, this could be organized in an interactive mode between the user and the planning system.

Third, an efficient data entry system can be developed that allows entering the data directly from standard RPM-LP and RPM-HA networks.

In this chapter, we have made some suggestions to overcome the limitations and practical problems to be encountered in the application of our model, the RPM network technique, and the RPMLPAS programming system. In short, they are: the implementation of a reporting and performance analysis function; the extension of the RPM network analysis to multiperiod models; the use of dual variables in the valuation of assets and the reporting of opportunities, and finally some recommendations regarding the improvement of the RPMLPAS programming system. Nevertheless, we have tried to develop a network language common for management science and information systems, in particular the accounting information subsystem.

BIBLIOGRAPHY

American Accounting Association 1971. Accounting and Information Systems. Committee on Information Systems. p. 287-350.

Charnes, A., Cooper, W., and Ijiri, Y. 1973. Breakeven Budgeting and Programming to Goals. Journal of Accounting Research. Spring: 16-43.

Charnes, A. and Cooper, W. 1967. Some Network Characterizations for Mathematical Programming and Accounting Approaches to Planning and Control. The Accounting Review: 24-52.

Charnes, A., Cooper, W., and Ijiri, Y. 1963. Breakeven Budgeting and Programming to Goals. Journal of Accounting Research. Spring: 16-43.

CODASYL Systems Committee 1971. Feature Analysis of Generalized Data Base Management Systems. Association of Computing Machinery.

D'Imperio 1969. Data Structures and their Representation in Storage. Pergamon Press, Headington Hill Hall, London. p. 1-75.

Grayson, C. 1973. Management Science and Business Practice. Harvard Business Review. July-August: 41-48.

Greynolds, E. 1972. The Accountants' Role in Operations Research. Bulletin of the ORSA. 20(1).

Hadley, G. 1962. Linear Programming. Addison-Wesley Co., Reading, Massachusetts.

Hartley, R. 1968. Operations Research and its Implications for the Accounting Profession. The Accounting Review. April: 321-332.

Holmgren, B., Obradovic, D., and Kolm, A. 1973. Remark on Algorithm 333 [H]; MINIT Algorithm for Linear Programming. Communication of the Association for Computing Machinery. 5:310.

Horngren, C. 1965. Accounting for Management Control: An Introduction. Prentice-Hall, Englewood Cliffs, N. J.

Ijiri, Y. 1965. Management Goals and Accounting for Control. North-Holland Publishing Company, Amsterdam.

Ijiri, Y. 1967. The Foundation of Accounting Measurement. Prentice-Hall, Englewood Cliffs, N. J.

Ijiri, Y., Kinard, J., and Putney, F. 1968. An Integrated Evaluation System for Budget Forecasting and Operating Performance. Journal of Accounting Research, Spring: 1-11.

Ijiri, Y., Levy, F., and Lyon, R. 1963. A Linear Programming Model for Budgeting and Financial Planning. Journal of Accounting Research, Autumn: 198-212.

Inoue, M. and Riggs, J. 1971. Describe Your System with Cause and Effect Diagrams. Industrial Engineering. 3:26-31.

Inoue, M. and Riggs, J. 1972. Resource Planning and Management Network: A Pedagogic Tool in Engineering Education. In: International Symposium of Systems Engineers, John E. Goldberg et al. eds., Lafayette, Indiana, Purdue University, October 23-27, 1972. p. 187-192.

International Business Machines. 1972. Communications Oriented Production Information and Control System. IBM G320-1981.

Keast, T. 1973. A Matrix Accounting Application to Financial Forecasting. 44th National Conference of ORSA (Nov. 12-14).

Kolm, A. and Dahlstrand 1971. Remark on Algorithm 333 [H]: MINIT Algorithm for Linear Programming. Communications of the Association for Computing Machinery. 1:50.

Chen, K. 1973. Application of Decomposition and Condensation Algorithms to the Logical Design of RPM networks. Oregon State University. Masters Dissertation.

Llewelyn, R. 1964. Linear Programming. Holt, Rinehart and Winston, New York. 207-220.

Messham, D. 1969. Remark on Algorithm 333 [H]: MINIT Algorithm for Linear Programming. Communications of the Association for Computing Machinery. 7:408.

Murphy, G. 1970. Algebraic Double Entry. The Accounting Review. April: p. 366-369.

Nicol, R. 1968. The Accounting Equation Revisited: A Conceptual Accounting Model. The Accounting Review. October: p. 777-779.

Obradovic, D. 1973. Remark on Algorithm 333 [H]: MINIT Algorithm for Linear Programming. 5:310.

Riggs, J. and Inoue, M. 1975. Introduction to Operations Research and Management Science. McGraw Hill, New York.

Rodolfo, C. and Sen, S. 1968. MINIT Algorithm for Linear Programming. Communications of the Association for Computing Machinery. June: p. 437-440.

Taha, H. 1971. Operations Research: An Introduction. Macmillan Company.

Urban, G. 1974. Building Models for Decision Makers. Interfaces of The Institute of Management Science. 4(3): 1-11.

Vazsonyi, A. 1973. Information Systems in Management Science. Interfaces of the Institute of Management Science. 3(3): 30-33.

APPENDIX I

Information Related Theories

Most of the existing information-related theories formalize different elements and aspects of the information subsystem and of the operating subsystem in a decision making context. The relationship with the service subsystem (e.g., engineering, research ...) is usually excluded from the analysis, despite the accounting practice of allocating the service costs to the several product cost centers. These theories can be grouped under Information Economics, Measurement Theory, Communication Theory, and Cybernetics and Control Theory (American Accounting Association, 1971). As shown in Figure A, the total decision system can be viewed as consisting of an environment, an information system, goal, resources, and a set of possible actions. Table A gives a summary of the different areas of focus for the different theories with respect to the total system of Figure A.

Table A. Areas of concentration of the four information theories.

Information Economics	1, 2, 3, 4, 5, 6, 7.
Measurement Theory	8 and 12.
Communication Theory	Data transmission for (10, 11), (11, 12), (12, 13), (11, 14).
Cybernetics and Control Theory	1, 2, 3, 4, 5, 6, 7, 8, 9.

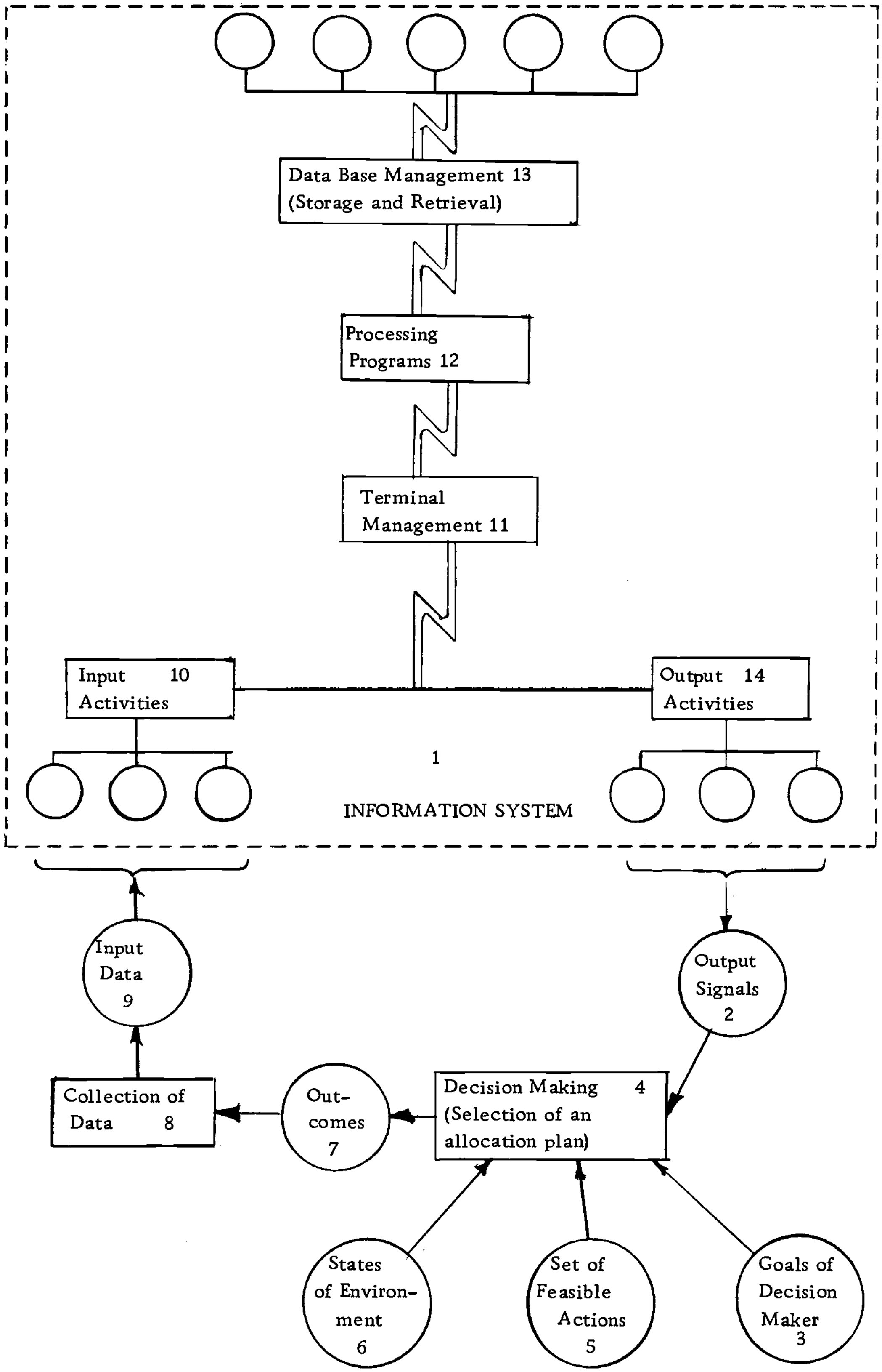


Figure A. The Information and Decision Subsystems.

This dissertation is not directly related to the measurement and communication theories and therefore these theories will not be discussed beyond this paragraph. Measurement theory evolves around assignment of numbers to objects and events. A distinction is usually made between the "fundamental measurement process", as part of the data collection processes and the "derived measurement process", as part of the data processing function. The former originates by direct observation of events and objects, while the latter is based on fundamental and/or derived measurements. Communication theory (also called "Statistical Information Theory") studies the transmission of data by studying the conversion procedures for encoding and decoding, as well as the characteristics of the transmission channel. The entropy measure of information is a direct outgrowth of this statistical information theory.

Information Economics

Information economics and cybernetics involve more or less the same elements from Figure 1.4. However, both theories stress different functions of the information involved. Information economics stands closer to the planning function by optimizing the selection of an information system in relation to the selection of an optimal decision (i.e., choosing the optimal alternative from a set of actions).

On the other hand, cybernetics focuses on the control function by concentrating on the selection of optimal feedback information needed for an optimal reaction to outcomes.

Information economics can be considered as an outgrowth of the more general decision theory. The most basic information issue that decision theory introduces is the value of perfect information. However, the information subsystem itself is not explicitly included in the decision making context. Figure B, derived from Figure A, shows the aspects that decision theory involves with respect to the total system.

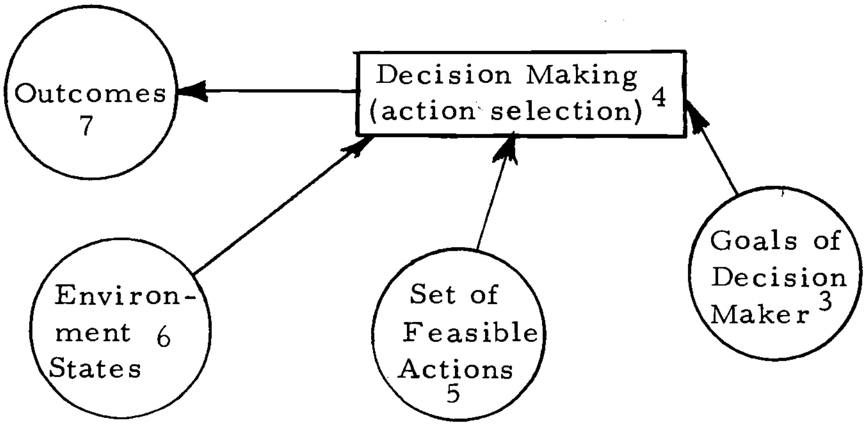


Figure B. The area of concentration of decision theory.

Unlike decision theory, the basic issue of information economics is the evaluation of additional information versus the cost incurred. The existing information together with additional information is considered to be another information system, meaning that information economics explicitly recognizes the alternative information system for a decision problem. Figure C, also derived from Figure A, shows the elements that information economics involves.

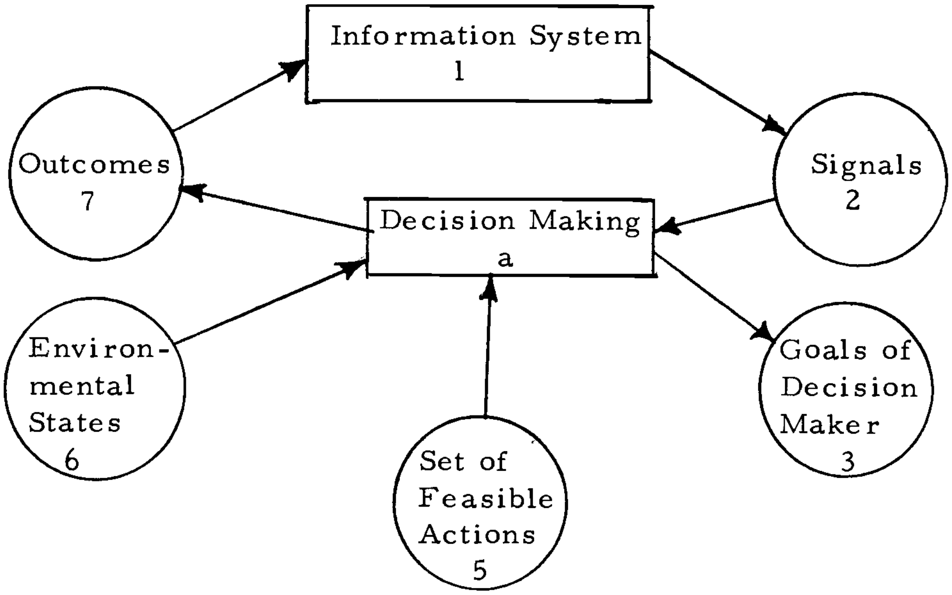


Figure C. The area of focus of information economics.

In the context of an LP model, it is usually assumed that parameter predictions (the signals) are deterministic in order to simplify the computation of solutions. In order to account for the uncertainty aspect of the parameters, sensitivity analysis can be used to answer the same questions that information economics formally addresses. If a small error in the predicted magnitude of a parameter has a significant impact on the optimal payoff value, and if it is possible to obtain additional information, which will improve the parameter prediction, that information should be obtained. In practice, additional information should be obtained if the prediction is critical and the cost of that information is within reason. For example, it is customary to obtain the demand forecast as accurately as possible because of its influence on production planning.

Because the actions of a decision maker are intended to be evaluatable, the payoff of the possible outcomes from actions has traditionally been measured in monetary terms. However, developments in investment theory (another branch of the general decision theory) recognizes "utility" as a superior measurement to monetary measures. It is beyond the scope of this appendix to review the utility measures and how utility measures and other behavioral implications should be included in the payoff function.

To conclude, research in information economics has provided a number of useful insights into the issues and problems of coupling the information and decision subsystems (Feltham, 1968, 1970, 1972; Demski, 1970, 1973; Marshak and Radner, 1972). This dissertation is intended to be an operational application of these theories with respect to a LP decision model and the cost information subsystem.

Cybernetics

Cybernetics covers the broad field of systems theory, and hence includes almost everything discussed in this appendix. In this section, however, we will only highlight some of the most important concepts of cybernetics relevant to the decision making environment.

This includes the use of feedback information on outcomes of actions, in order to determine reactions that should be given in order to accomplish the desired goals of the system.

The control function of decision making system is a first order feedback loop. Observations about events and objects are assembled in the information system, and reports of the actual state are made available to the decision maker, who compares these reports with his goals or standards. This will prompt him to corrective action if he sees any benefit of this effort (see Figure D). This basic first order schema can be automated by letting the information system take over the comparison of standards with actual results. The standard cost accounting system is an example of a first order feedback system.

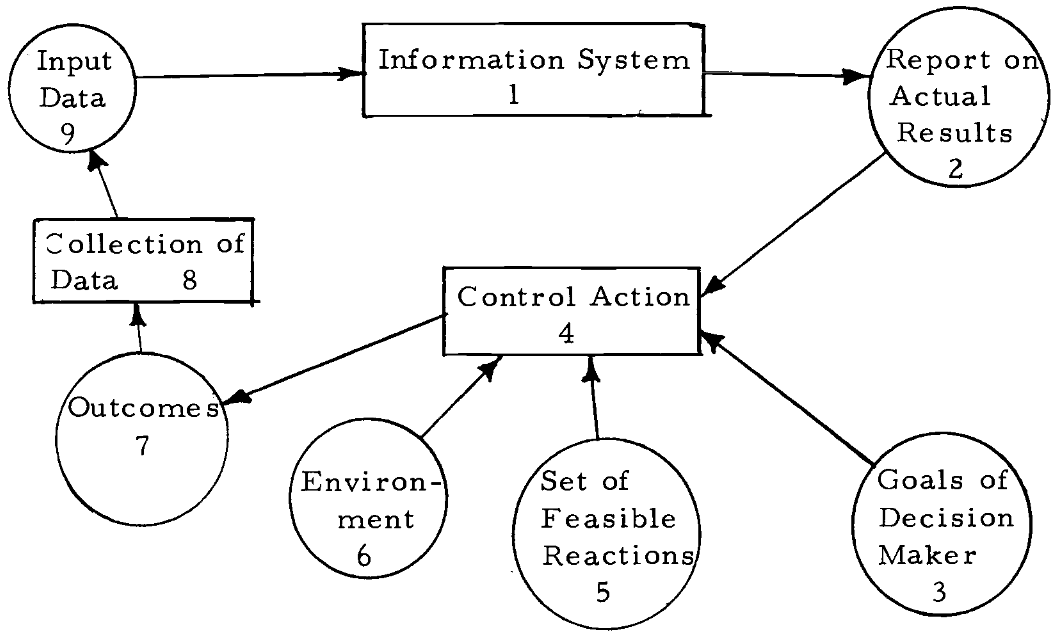


Figure D. The control function as a first-order feedback loop.

The time element is not shown in Figure D but is an important factor in obtaining stability for the system. Furthermore, by giving a memory to the system through monitoring changes in the environment, it becomes possible to take appropriate action in advance. An example of a monitoring and memory system is the CUSUM chart used in statistical quality control.

Cybernetics goes further and gives consideration to systems that can learn, and thus develops new techniques for handling new situations, rather than applying the initially established decision rules. At the highest level, consideration is given to systems that can modify their goals as a result of what they have learned. The study of these higher order systems leads to developments in the domain of artificial intelligence.

To conclude, we have tried to discuss the integration between the information system and the decision model as studied by the information related theories, and to indicate their relationship to our study. In doing so, we have simplified to a great extent this complex and sophisticated body of knowledge.

APPENDIX II

Primal-dual Data Structure for Model Data of Example One

POINTER ADDRESS	RESOURCE NODE INDEX	ACTIVITY NODE INDEX	TECHNOLOGICAL COEFFICIENT	PRIMAL LIST POINTERS	DUAL LIST POINTERS
1	1	4	-1.000	15	0
2	2	1	8.000	7	3
3	2	2	5.000	8	4
4	2	3	3.000	9	5
5	2	5	1.000	0	6
6	2	9	100.000	11	0
7	3	1	6.000	12	8
8	3	2	9.000	13	9
9	3	3	8.000	14	10
10	3	6	1.000	0	11
11	3	9	200.000	0	0
12	4	1	-7.000	17	13
13	4	2	-6.000	18	14
14	4	3	-8.000	19	15
15	4	4	4.000	20	16
16	4	7	1.000	0	0
17	5	1	-5.000	0	18
18	5	2	-9.000	0	19
19	5	3	-4.000	0	20
20	5	4	3.000	0	21
21	5	8	1.000	0	0
22	0	0	0	23	0
23	0	0	0	24	0
24	0	0	0	25	0
25	0	0	0	26	0
26	0	0	0	27	0
27	0	0	0	28	0
28	0	0	0	29	0
29	0	0	0	30	0
30	0	0	0	31	0
31	0	0	0	32	0
32	0	0	0	33	0
33	0	0	0	34	0
34	0	0	0	35	0
35	0	0	0	36	0
36	0	0	0	0	0

ACTIVITY POINTER SEGMENT

ACTIVITY NODE 1	POINTER = 2
ACTIVITY NODE 2	POINTER = 3
ACTIVITY NODE 3	POINTER = 4
ACTIVITY NODE 4	POINTER = 1
ACTIVITY NODE 5	POINTER = 5
ACTIVITY NODE 6	POINTER = 10
ACTIVITY NODE 7	POINTER = 16
ACTIVITY NODE 8	POINTER = 21
ACTIVITY NODE 9	POINTER = 6

RESOURCE POINTER SEGMENT

RESOURCE NODE 1	POINTER = 1
RESOURCE NODE 2	POINTER = 2
RESOURCE NODE 3	POINTER = 7
RESOURCE NODE 4	POINTER = 12
RESOURCE NODE 5	POINTER = 17

APPENDIX III

Users' Information on the RPMLPAS Programming System

The RPM-LP-AS system is designed to be general in the solution of the LP model, the interfacing with an accounting system, and the updating of accounts. This means that by entering the proper set of control cards and data, the solution for the LP model and the information for the accounting system can be computed as needed for the financial statements. However, RPM-LP-AS does not provide a report generating capacity that allows a generalization of report writing. This means that, in actual applications, the subroutine FINSTAT may have to be modified so that computations and report formats conform to the needs of the particular organization.

In the following pages, we shall take a closer look at the preparation of the input data by dividing this total input data deck (or file with card images) into four sections:

- Control Cards
- LP Data Matrix
- Interface Data
- Accounting System Data

Table D. illustrates the input data for example two.

Control Cards

All control variables use the format I3.

A. LP Control Card.

Variable Name	Column Location	Variable Description
M	1-3	Number of equations.
IP	4-6	Number of equality constraints.
N	7-9	Number of activities.
LCOL	10-12	Number of columns in LP matrix.
ITST	13-15	Iteration number after which complete print out of each iteration starts.
MIT	16-18	Maximum number of iterations (saveguard against cycling).

B. Interface Control Card.

LRCM	1-3	Last row of conversion matrix.
LPTR	4-6	Number of transactions generated from LP solution.
LTR	7-9	Total number of transactions.

C. Accounting System Control Card

MAC	1-3	Number of rows in account matrix.
NAC	4-6	Number of columns in account matrix.

LP Data Matrix

The initial model data is entered with the format 8F10. 3; one row may occupy several input cards. The general format of the initial tableau before packing is given in Table B (Salazar and Sen, 1968). The tolerance is the same for all matrix coefficients and is $TD = 10^{-8}$. If a coefficient is greater than or equal to -TD, it is considered to be nonnegative. The first condition of optimality is the nonnegativity of the first row except for its last element that contains the objective function value and may be negative. The second condition is the nonnegativity of the last column, except necessarily for its first element.

Table B. Format for the LP data matrix.[a]

	1	. . . .	LCOL	
1	$-c_j$ (1*N)	0 (1*M-IP)	0	
. . . .	A(M*N)	Identity Matrix (M-IP*M-IP)	b (M*1)	(M-IP) Inequality Constraints
. . M+1		Zero Matrix (IP*M-IP)		IP Equality Constraints

[a] N= number of variables. IP = number of equality constraints.
M = number of constraints LCOL = total number of columns.

Interface Data

The interface data consists of conversion matrix information and the transaction array.

Conversion Matrix Information

As indicated in Chapter V, the conversion matrix uses data elements available through the LP model data. The conversion matrix information consists of three parallel data arrays:

NTR (100)	for the transaction number of the transaction (TR(100)) to be updated.
NROW(100)	for the resource node number of the initial model data corresponding to desired coefficient.
NVAR(100)	for the activity number of the initial model data corresponding to desired coefficient, and the variable of the LP solution to be multiplied with that coefficient.

NROW and NVAR indicate the indices used by the function XTCHC to access the technological coefficient in the linked list structure. In case the sign of that coefficient has to be changed, a minus sign is placed in front of the NROW value. Also, in case the LP solution value is the same as the transaction value (corresponds to a coefficient one in the conversion matrix), the NROW value is set equal to zero, and no search is executed for a coefficient.

Transaction Information

The transaction information consists of three parallel arrays:

TR (100) for the transaction value

ICR (100) for the lowest level account to be credited.

IDR (100) for the lowest level account to be debited.

This information is assumed to be ordered into two groups: the first LPTR transactions will be generated from the LP model solution, while the last LTR-LPTR transactions are budgeted transactions beyond the decision domain of the LP model, but that nevertheless help determine the end-of-period outlook of the financial statements.

Accounting System Data

The hierarchical structure of the accounting system is communicated through the account matrix, that is further followed in the data set by the beginning-of-period banalce value.

With each column of the account matrix corresponds an essential account, while each row corresponds to an higher level or aggregate account. An one value is located in each column to indicate the higher level accounts on which an essential account is dependent. The rows of the account matrix are ordered such that the first row corresponds to the first higher order, while the last rows correspond to the last or highest hierarchical levels. Such a matrix is illustrated for the accounts of the expanded incidence matrix in Table C.

This matrix is converted to a pointer array (PACB (100)) that runs parallel to the beginning and ending account balances (BACB(100) and EACB(100)). The subroutine UPDATE uses this pointer array to form loops while updating the account balances.

Table C. Account matrix for example two.

Aggregate Account \ Essential Account	Cash	Acc. Receivable	Inv. Materials	Inv. Fin. Prod.	Fixed Assets	Short T. Liab.	Long T. Liab.	Capital Stock	New Investment	Sales	Taxes	Dividends
Total Liabilities						1	1					
Oper. Income										1		
Retained Earnings										1	1	1
Owner's Equity								1	1	1	1	1
Total Equities						1	1	1	1	1	1	1
Total Inventory			1	1								
Liquid Assets	1	1										
Current Assets	1	1	1	1								
Total Assets	1	1	1	1	1							

Table D. Entry data for RPMLPAS (example two).

```
015008010018015015
011010013
009012
       0       0       0       0       0       0   -.250   -.325
       0       0       0       0       0       0       0       0
       0       0
   1.000       0       0       0       0       0       0       0
       0       0   1.000       0       0       0       0       0
       0  70.000
       0       0       0       0   1.000       0       0       0
       0       0       0   1.000       0       0       0       0
       0  17.500
       0   1.000       0       0       0       0       0       0
       0       0       0       0   1.000       0       0       0
       0  18.000
       0       0       0       0       0       0   2.000   2.000
       0       0       0       0       0   1.000       0       0
       0  22.000
       0       0       0       0       0       0       0   3.000
       0       0       0       0       0       0   1.000       0
       0  10.500
       0       0       0       0       0       0  -1.000       0
       0       0       0       0       0       0       0   1.000
       0  -6.000
       0       0       0       0       0       0       0  -1.000
       0       0       0       0       0       0       0       0
   1.000  -3.000
   1.000  -1.000    .400  -1.000   -.450       0       0       0
       0       0       0       0       0       0       0       0
       0   2.000
       0       0       0       0   1.000       0  -2.000  -2.000
       0       0       0       0       0       0       0       0
       0  -2.000
       0   1.000       0       0   -.550  -1.000       0       0
       0       0       0       0       0       0       0       0
       0       0
  -1.000       0    .600       0       0       0       0       0
       0       0       0       0       0       0       0       0
       0       0
       0       0       0  -1.000       0       0    .500   1,300
       0       0       0       0       0       0       0       0
       0       0
       0       0       0       0       0       0    .250    .325
  -1.000       0       0       0       0       0       0       0
       0       0
       0       0       0   1.000       0       0   2.000   2.000
       0  -1.000       0       0       0       0       0       0
       0       0
       0       0  -1.000       0       0       0       0       0
   1.000   1.000       0       0       0       0       0       0
       0       0
       1   -9    5
       2  -11    5
       3    5    7
       3    5    8
       4    0    1
       5    0    2
       6    0   10
       7    0    9
       8    9    3
       9   12    3
      10    0    4
   000001100000
   000000000100
   C00000000111
   000000011111
   000001111111
   C01100000000
   110000000000
   111100000000
   111110000000
               0    1    3
               0    6    3
               0    3    4
               0    2    1
               0    1    6
               0    4   10
               0   14   10
               0   10    1
               0   10    2
               0    1    4
         1.500      5   14
          .500      6   14
         3.000      7    5
         5.000
        20.000
         5.000
         2.5000
        10.000
       -18.000
        -7.500
             0
             0
             0
             0
             0
       -25.500
             0
             0
       -17.000
       -42.500
         7.500
        25.000
        32.500
        42.500
```

APPENDIX IV

RPMLPAS: Source Program

```
      DEFINE GLOBAL2
      INTEGER PPRL,PDUL,RESN,ACTN,PFR,PFA,PLAC,PACB
      COMMON/MINIT/IP,IP1,K,KM1,L,LM1,M,MP1,MP2,N,NP1,LCOL,LCM1,IM1,
     *JMIN1,ICT,GMIN,PHIMAX,THETA,GAMMA,DELTA,PHI,TO,ITST,MIT,IP2,IR1,
     *IND(50),IND1(50),THMIN(50),IMIN(50),
     *JMAX(30),DELMAX(30)
      COMMON/RPMLP/BUF(50),PFA(50),PLAC(50),
     *PFR(30),
     *PPRL(1000),PDUL(1000),TCHC(1000),RESN(1000),ACTN(1000)
     *,IAV,NL,LAST,LAST1,LAST2
      COMMON/INTERF/TR(75),NTR(100),NROW(100),NVAR(100),ICR(100),
     * IDR(100),
     * ITAPE,LRCM,LTR,MAC,NAC,MNAC,LPTR
      COMMON/RESULTS/JH(50),X(50),
     * ICHK(30),W(30)
      COMMON/ACCSYS/IAC(80),BACB(100),EACB(100),PACB(100),NLAHA(80)
      END


      PROGRAM RPMLPAS
CCCCC
CCCCC MAIN PROGRAM MAKES CALLS FOR EACH MAJOR STEP AS
CCCCC DESCRIBED IN FIGURE III.1. THE MAIN PROGRAM AND ITS
CCCCC 17 SUBROUTINES USE 3,971 WORDS, WHILE THE COMMON DATA
CCCCC AREA OCCUPIES 8,166 WORDS.THESE ROUTINES AREPROGRAMMED
CCCCC UNDER THE OS-3 OPERATING SYSTEM OF OREGON STATE
CCCCC UNIVERSITY USING THE CDC-3300 COMPUTER.
CCCCC
      INCLUDE GLOBAL2
      IR1=5
      IP1=1
      ITAPE=6
      LPTST=0
      TO=10.E-8
      ICT=0
CCC
CCC READ CONTROL CARDS
CCC
      READ(IR1,1)M,IP,N,LCOL,ITST,MIT
1     FORMAT(6I3)
      MP1=M+1
      MP2=M-IP+2
      NP1=N+1
      LCM1=LCOL-1
      READ(IR1,1)LRCM,LPTR,LTR
      READ(IR1,1)MAC,NAC
CCC
CCC READ LP MODEL DATA AND PACK
CCC
```

```
      CALL PACK
CCC
CCC READ IN INTERFACE DATA AND ACCOUNTING SYSTEM DATA
CCC
      CALL INTACC
CCC
CCC STORE SYSTEM ON TAPE
CCC
      CALL STORE
CCC
CCC EXECUTE LP MODEL
CCC
      CALL RPMLP
CCC
CCC FIND INITIAL SYSTEM DATA BACK AND CONVERT LP RESULT TO TRANSACTION A
CCC
      CALL CONVERT
CCC
CCC UPDATE ACCOUNT BALANCES.
CCC
      CALL UPDATE
CCC
CCC PRINT FINANCIAL STATEMENTS.
CCC
      CALL FINSTAT
      STOP
      END


      SUBROUTINE INTACC
CCCCC
CCCCC READS IN INTERFACE DATA AND ACCOUNTING SYSTEM DATA (I.E. HIERAR-
CCCCC CHICAL DATA STRUCTURE, TRANSACTION DATA AND BEGINNING BALANCE
CCCCC ACCOUNTS.
CCCCC
      INCLUDE GLOBAL2
CCC
CCC READ IN CONVERSION DATA
CCC
      DO 15 I=1,LRCM
15    READ(IR1,16)NTR(I),NROW(I),NVAR(I)
16    FORMAT(3I5)
CCC
CCC READ IN HIERARCHICAL ACCOUNTING DATA AND TRANSFORM TO ACCOUNT
CCC DATA STRUCTURE.
CCC
      DO 5 I=1,MAC
      READ(IR1,1)(IAC(J),J=1,NAC)
1     FORMAT(80I1)
      DO 4 J=1,NAC
      IF(IAC(J).EQ.0) GOTO 4
      ITMP=NLAHA(J)
      IF(ITMP.EQ.0)ITMP=J
      INAC=I+NAC
      PACB(ITMP)=INAC
```

```
      NLAHA(J)=INAC
4     CONTINUE
5     CONTINUE
CCC
CCC READ IN TRANSACTION DATA
CCC
      DO 10 I=1,LTR
10    READ(IR1,11)TR(I),ICR(I),IDR(I)
11    FORMAT(F10.3,2I5)
CCC
CCC READ IN BEGINNING BALANCE OF ACCOUNTS.
CCC
      MNAC=NAC+MAC
      DO 6 I=1,MNAC
      READ(IR1,12)BACB(I)
12    FORMAT(F10.3)
C TEST
6     WRITE(IP1,7)I,PACB(I),BACB(I)
7     FORMAT(2I5,F10.3)
      RETURN
      END



      SUBROUTINE PACK
      INCLUDE GLOBAL2
CCCCC
CCCCC READ IN LP DATA AND STORE TECH. COEFF. IN A LINKED LIST
CCCCC
      IAV=1
      NL=MP1*LCOL
      IF(NL.GT.1000)NL=1000
CC SET UP AVAILABLE LIST
      DO 240 I=1,NL
240   PPRL(I)=I+1
      PPRL(NL)=0
CC READ ROW
      DO 241 I=1,MP1
      READ(IR1,51)(BUF(J),J=1,LCOL)
51    FORMAT(8F10.3)
CC START PACKING
      DO 242 J=1,LCOL
      IF(BUF(J).EQ.0.) GOTO 242
CC POINTERS OUTSIDE MATRIX
      IF(PFR(I).EQ.0)PFR(I)=IAV
      IF(PFA(J).EQ.0)PFA(J)=PLAC(J)=IAV
CC ROW NUMBER,COLUMN NUMBER, TECH. COEFF.
      RESN(IAV)=I
      ACTN(IAV)=J
      TCHC(IAV)=BUF(J)
CC STORE POINTERS FOR DUAL AND PRIMAL LISTS
      ITMP1=PPRL(IAV)
      PDUL(IAV)=ITMP1
      ITMP1=IAV
      IAV=PPRL(IAV)
CC CHECK IF FREE STORAGE SPACE IS EXCEEDED
```

```
      IF(IAV.LE.NL) GOTO 244
      WRITE(IR1,245)
245   FORMAT(1X,≠NOT ENOUGH FREE STORAGE SPACE≠,/,
     *1X,≠ENLARGE NL IN THE MAIN PROGRAM≠)
      CALL EXIT
244   IF(PLAC(J).EQ.ITMP1) GOTO 242
      ITPL=PLAC(J)
      PPRL(ITPL)=ITMP1
      PLAC(J)=ITMP1
242   CONTINUE
CC ZERO LAST ROW AND COLUMN ELEMENTS
      POUL(ITMP1)=0
241   CONTINUE
      DO 243 J=1,LCOL
      ITPL=PLAC(J)
243   PPRL(ITPL)=0
      RETURN
      END


      SUBROUTINE STORE
CCCCC
CCCCC STORES THE INITIAL DATA
CCCCC
      INCLUDE GLOBAL2
      REWIND ITAPE
      ENDFILE ITAPE
      BUFFEROUT (ITAPE,0)(IF,DELMAX(30))
11    GOTO(11,12,13,14),UNITSTF(ITAPE)
12    BUFFER OUT (ITAPE,0)(BUF(1),LAST2)
21    GOTO(21,22,13,14),UNITSTF(ITAPE)
22    ENDFILE ITAPE
      RETURN
13    WRITE(IP1,20)
20    FORMAT(1X,≠END OF FILE ENCOUNTERED≠)
      REWIND ITAPE
      CALL EXIT
14    WRITE(IP1,26)
26    FORMAT(1X,≠TAPE I/O ERROR≠)
      REWIND ITAPE
      CALL EXIT
      END


      SUBROUTINE CONVERT
CCCCC
CCCCC CONVERTS LP RESULTS INTO TRANSACTION AMOUNTS
CCCCC
      INCLUDE GLOBAL2
      WRITE(IP1,37)
37    FORMAT(//,1X,≠CONVERT ENTERED≠,//)
      REWIND ITAPE
      CALL SEFF(ITAPE)
      BUFFER IN (ITAPE,0) (IP,DELMAX(30))
```

```
11     GOTO(11,12,13,14),UNITSTF(ITAPE)
12     BUFFER IN(ITAPE,0)(BUF(1),LAST2)
21     GOTO(21,22,13,14),UNITSTF(ITAPE)
22     CONTINUE
       DO 25 I=1,LPTR
25     TR(I)=0.
CCC
CCC EACH LOOP IN DO LOOP 30 FETCHES ONE COEFFICIENT AND
CCC UPDATES THE TRANSACTION VALUE.
CCC
       DO 30 I=1,LRCM
       ITMP1=NTR(I)
       ITMP2=NROW(I)
       ITMP3=NVAR(I)
       WRITE(IP1,35)ITMP1,ITMP2,ITMP3
35     FORMAT(1X,≠ITMP VALUES≠,3I5)
CCC
CCC IF ROW COEFF. IS NEGATIVE, CHANGE THE SIGN OF THE
CCC COEFF. BEFORE MULTIPLYING WITH THE VARIABLE.
CCC
       IF(ITMP2)28,31,29
28     ITMP2=-ITMP2
       CC=-XTCHC(ITMP2,ITMP3)
       GOTO 32
29     CC=XTCHC(ITMP2,ITMP3)
32     TR(ITMP1)=TR(ITMP1)+X(ITMP3)*CC
       GOTO 30
31     TR(ITMP1)=TR(ITMP1)+X(ITMP3)
30     WRITE(IP1,34)ITMP1,TR(ITMP1),CC
34     FORMAT(1X,I5,2F10.4)
       RETURN
13     WRITE(IP1,26)
26     FORMAT(1X,≠END OF FILE ENCOUNTERD≠)
       REWIND ITAPE
       CALL EXIT
14     WRITE(IP1,27)
27     FORMAT(1X,≠TAPE I/O ERROR≠)
       REWIND ITAPE
       CALL EXIT
       END


       SUBROUTINE UPDATE
CCCCC
CCCCC UPDATES LOWER AND HIGHER LEVEL ACCOUNTS FOR TRANSACTIONS
CCCCC
       INCLUDE GLOBAL2
CCC
CCC SET END-OF-PERIOD BALANCE EQUAL TO BEG.-OF-PER. BAL.
CCC
       DO 5 I=1,MNAC
5      EACB(I)=BACB(I)
CCC
CCC EACH LOOP IN DO LOOP 100 UPDATES THE LOWER AND HIGHER LEVEL
CCC ACCOUNTS RELATED TO TRANSACTION I
```

```
CCC
      DO 100  I=1,LTR
      ITMPCC=ITMPC=ICR(I)
      ITMPDD=ITMPD=IDR(I)
      IF(ITMPC)30,99,25
30    ITMPCC=ITMPC=-ITMPC
      WRITE(IP1,65)EACB(ITMPCC),EACB(ITMPDD)
65     FORMAT(2F10.3)
      EACB(ITMPC)=EACB(ITMPC)-TR(I)
      EACB(ITMPD)=EACB(ITMPD)+TR(I)
13    ITMPC=PACB(ITMPC)
      IF(ITMPC.EQ.0)  GOTO 99
      IF(ITMPC-ITMPD)10,11
10    EACB(ITMPC)=EACB(ITMPC)-TR(I)
      GOTO 13
11    ITMPD=PACB(ITMPD)
      IF(ITMPD.EQ.0)  GOTO 100
      EACB(ITMPD)=EACB(ITMPD)+TR(I)
      GOTO 11
25    IF(ITMPD)34,99,50
CCC
CCC THE ACCOUNT TO BE DEBETED IS HIERARCHICAL LOWER THAN THE ONE TO BE
CCC CREDITED.
CCC
34    ITMPDD=ITMPD=-ITMPD
      WRITE(IP1,65)EACB(ITMPCC),EACB(ITMPDD)
      EACB(ITMPC)=EACB(ITMPC)-TR(I)
      EACB(ITMPD)=EACB(ITMPD)+TR(I)
28    ITMPD=PACB(ITMPD)
      IF(ITMPD.EQ.0)  GOTO 99
      IF(ITMPD-ITMPC)26,27
26    EACB(ITMPD)=EACB(ITMPD)+TR(I)
      GOTO 28
27    ITMPC=PACB(ITMPC)
      IF(ITMPC.EQ.0)  GOTO 100
      EACB(ITMPC)=EACB(ITMPC)-TR(I)
      GOTO 27
CCC
CCC NORMAL CASE. BOTH ACCOUNTS ARE ADVANCED IF APPROPRIATE.
CCC
50    WRITE(IP1,65)EACB(ITMPCC),EACB(ITMPDD)
      EACB(ITMPC)=EACB(ITMPC)-TR(I)
      EACB(ITMPD)=EACB(ITMPD)+TR(I)
54    ITMPD=PACB(ITMPD)
51    ITMPC=PACB(ITMPC)
52    IF(ITMPD.NE.0)  GOTO 53
      IF(ITMPC.EQ.0)  GOTO 100
      EACB(ITMPC)=EACB(ITMPC)-TR(I)
      GOTO 51
53    IF(ITMPC.EQ.ITMPD)  GOTO 100
      EACB(ITMPD)=EACB(ITMPD)+TR(I)
      IF(ITMPC.EQ.0)ITMPD=PACB(ITMPD)
      IF(ITMPC.EQ.0)  GOTO 52
      EACB(ITMPC)=EACB(ITMPC)-TR(I)
      GOTO 54
100   WRITE(IP1,21)TR(I),ICR(I),IDR(I),BACB(ITMPCC),EACB(ITMPCC),
```

```
     * BACB(ITMPOD),EACB(ITMPOD)
21    FORMAT(F10.3,2I5,4F10.3)
C TEST
      DO 110 I=1,MNAC
110   WRITE(IP1,111)I,BACB(I),EACB(I)
111   FORMAT(I5,2F10.3)
      RETURN
CCC
CCC ERROR STATEMENT
CCC
99    WRITE(IP1,98)
98    FORMAT(1X,≠ERROR IN ACCOUNTING SYSTEM≠)
      CALL EXIT
      END



      SUBROUTINE FINSTAT
CCCCC
CCCCC PRINTS THREE MAIN STATEMENTS ... BALANCE SHEET (WITH BEGINNING AND
CCCCC ENDING BALANCES), BUDGETED INCOME STATEMENT AND BUDGETED STATEMENT
CCCCC OF FUNDS FLOW.
CCCCC
      INCLUDE GLOBAL2
CCC
CCC EXPRESS VALUES IN DOLLARS INSTEAD OF THOUSAND OF DOLLARS
CCC CHANGE ALL NEGATIVE VALUES TO POSITIVE ONES
CCC
      DO 1 I=1,MNAC
      BACB(I)=BACB(I)*1000.
      EACB(I)=EACB(I)*1000.
      IF(BACB(I).GT.0.) GOTO 1
      BACB(I)=-BACB(I)
      EACB(I)=-EACB(I)
1     CONTINUE
CCC
CCC ONLY FOR DISSERTATION.
CCC WRITE ... TABLE 4.5: FINANCIAL STATEMENTS
CCC
      WRITE(IP1,30)
30    FORMAT(/////,7X,≠TABLE 4.5  FINANCIAL STATEMENTS≠)
CCC
CCC BALANCE SHEET
CCC
      WRITE(IP1,10)
10    FORMAT(1H1,/////,60X,≠BALANCE  SHEET≠)
      WRITE(IP1,11)BACB(6),EACB(6),BACB(1),EACB(1),BACB(7),EACB(7),
     * BACB(2),EACB(2),BACB(13),EACB(13),BACB(19),EACB(19)
11    FORMAT(//,45X,≠BEGINNING≠,5X,≠ENDING≠,36X,≠BEGINNING≠,5X,≠ENDING≠,
     */45X,≠BALANCE≠,6X,≠BALANCE≠,36X,≠BALANCE≠,6X,≠BALANCE≠,
     */,7X,≠ASSETS≠,58X,≠EQUITIES≠,
     */9X,≠CURRENT ASSETS≠,50X,≠LIABILITIES≠,
     */11X,≠LIQUID ASSETS≠,51X,≠ACCOUNTS PAYABLE≠,8X,2(F10.2,2X),
     */13X,≠CASH≠,26X,2(F10.2,2X),8X,≠LONG TERM LIABILITIES≠,3X,
     *2(F10.2,2X),
     */,13X,≠ACCOUNTS RECEIVABLE≠,11X,2(F10.2,2X),8X,≠TOTAL LIABILITIES≠
```

```
     *,7X,2(F10.2,2X),
     */,13X,≠TOTAL LIQUID ASSETS≠,11X,2(F10.2,2X))
      WRITE(IP1,12)BACB(16),EACB(16),BACB(3),EACB(3),BACB(4),EACB(4),
     *BACB(18),EACB(18),BACB(20),EACB(20),BACB(5),EACB(5),BACB(21),
     *EACB(21),BACB(17),EACB(17)
12    FORMAT(73X,≠EQUITY≠,20X,2(F10.2,2X),
     */,11X,≠INVENTORIES≠,
     */,13X,≠RAW MATERIALS≠,17X,2(F10.2,2X),
     */13X,≠FINISHED PRODUCT≠,14X,2(F10.2,2X),
     */,13X,≠TOTAL INVENTORIES≠,13X,2(F10.2,2X),
     *//,11X,≠TOTAL CURRENT ASSETS≠,12X,2(F10.2,2X),
     *//,9X,≠FIXED ASETS (LESS DEPRECIATION)≠,3X,2(F10.2,2X),
     */,43X,≠----------≠,2X,≠----------≠,34X,≠----------≠,2X,
     *≠----------≠,
     */,9X,≠TOTAL ASSETS≠,22X,2(F10.2,2X),4X,≠TOTAL EQUITIES≠,14X,
     *2(F10.2,2X),
     */43X,≠==========≠,2X,≠==========≠,34X,≠==========≠,2X,
     *≠==========≠,//////)
CCC
CCC BUDGETED INCOME STATEMENT
CCC BUDGETED STATEMENT OF FUNDS FLOW
CCC
      DO 25 I=1,LTR
25    TR(I)=TR(I)*1000.
      SALES=TR(8)+TR(9)
      GROSPR=SALES-TR(6)
      GROSINC=EACB(14)+TR(11)
      TOTSRC=GROSINC+TR(13)
      WCI=TOTSRC-TR(13)
      WRITE(IP1,22)EACB(14),TR(11),SALES,TR(6),GRCSINC
22    FORMAT(77X,≠BUDGETED STATEMENT OF FUNDS FLOW≠
     *//,22X,≠BUDGETED INCOME STATEMENT≠,24X,≠INCOME FROM OPERATIONS≠,
     */,80X≠NET INCOME≠,9X,F10.2,
     */80X,≠DEPRECIATION≠,7X,F10.2,
     */,7X,≠SALES≠,43X,F10.2,34X,≠----------≠,
     */,7X,≠COST FINISHED PRODUCTS SOLD≠,21X,F10.2,15X,≠GROSS INCOME≠,
     *19X,F10.2)
      WRITE(IP1,23)GROSPR,TR(13),TR(12),TOTSRC,TR(11),TR(13),EACB(14),
     *WCI
23    FORMAT(55X,≠----------≠,
     */,7X,≠GROSS PROFIT≠,36X,F10.2,6X,≠INCOME FROM OTHER SOURCES≠,15X,
     *F10.2,
     */7X,≠FIXED COST≠,38X,F10.2,6X,≠TOTAL SOURCES≠,27X,F10.2,
     */7X,≠DEPRECIATION≠,36X,F10.2,6X,≠APPLICATION OF FUNDS≠,20X,F10.2,
     */,55X,≠----------≠,46X,≠----------≠,
     */7X,≠NET INCOME≠,38X,F10.2,6X,≠INCREASE IN WORKING CAPITAL≠,13X,
     * F10.2,
     */,55X,≠==========≠,46X,≠==========≠)
      RETURN
      END


      SUBROUTINE RPMLP
CCCCC
CCCCC THIS ROUTINE SOLVES THE LP MODEL ACCORDING TO THE
```

```
CCCCC MINIT ALGORITHM PROPOSED BY LLEWELYN 81964). THIS
CCCCC ROUTINE AND ITS 10 SUBROUTINES USE 2,562 WORDS
CCCCC OF CORE MEMORY.
CCCCC
      INCLUDE GLOBAL2
CCC
CCC PRINT OUT TABLEAU IN STANDARD FORM BY UNPACKING LINKED LIST
CCC
      CALL LPTEST(IM,JMIN)
CCC
CCC IF THERE ARE ANY EQUALITY CONSTRAINTS IN THE MODEL, THE PROGRAM
CCC FIRST JUMPS TO PHASE ONE
CCC
      IF(IP.NE.0)CALL PHASE1
CCC
CCC START PHASE 2
CCC
CC IND(L) KEEPS TRACK OF COLUMNS WITH NEGATIVE CJ-ZJ
2     L=1
      K=1
      LAST=PFR(1)
      IF(LAST)500,504
500   IF(TCHC(LAST).GE.-TD) GOTO 3
      IND(L)=ACTN(LAST)
      L=L+1
3     IF(PDUL(LAST).EQ.0) GOTO 504
      LAST=PDUL(LAST)
      GOTO 500
CC IND1(K) KEEPS TRACK OF ROWS WITH NEGATIVE BJ VALUE
504   LAST=PFA(LCOL)
      IF(LAST)505,506
505   IF(TCHC(LAST).GE.-TD) GOTO 4
      IND1(K)=RESN(LAST)
      K=K+1
4     IF(PPRL(LAST).EQ.0) GOTO 506
      LAST=PPRL(LAST)
      GOTO 505
CC CHECK IF THE SOLUTION IS PRIMAL OPTIMAL (L=1 AND K=1)
506   IF(L.GT.1) GOTO 5
      IF(K.EQ.1) CALL RESULTS
      IF(K.EQ.1) RETURN
CCC
CCC SOLUTION IS PRIMAL BUT INFEASIBLE
CCC  CHECK IF DUAL IS UNBOUNDED
CCC
      IF(K.GT.2)GOTO 70
      II=IND1(1)
      LAST=PFR(II)
      IF(LAST)520,521
520   IF(TCHC(LAST).LT.-TD) GOTO 70
      IF(PDUL(LAST))525,521
525   LAST=PDUL(LAST)
      GOTO 520
CCC
CCC PRIMAL SOLUTION IS NOT FEASIBLE
CCC
```

```
521    WRITE(IP1,9)
9      FORMAT(1X,≠PRIMAL PRCELEM HAS NO FEASIELE SCLUTICN≠,/,
      *1X,≠DUAL OBJECTIVE FUNCTION IS UNBOUNDED≠)
       GOTO 99
CC CHECK IF THE SOLUTION IS PRIMAL FEASIBLE. IF SO NO MORE DUAL-TYPE
CC ITERATIONS TO BE EXECUTED.
 5     IF(L.GT.2) GOTO 10
       IF(K.GT.1) GOTO 72
       JJ=IND(1)
       LAST=PFA(JJ)
       IF(LAST)530,531
530    IF(TCHC(LAST).GT.TD) GOTO 71
       IF(PPRL(LAST))535,531
535    LAST=PPRL(LAST)
       GOTO 530
531    WRITE(IP1,13)
CCC
CCC PRIMAL SOLUTION IS UNBOUNDED
CCC
       WRITE(IP1,13)
13     FORMAT(1X,≠PRIMAL OBJECTIVE FUNCTION IS UNBOUNDED≠,/,1X,
      *≠DUAL PROBLEM HAS NO FEASIBLE SCLUTION≠)
       GOTO 99
10     IF(K.EQ.1)GOTO 71
       GOTO 72
CC MAKE A DUAL-TYPE ITERATICN
70     CALL RPMPHI(IMAX,JM)
       CALL ROWTRANS(IMAX,JM)
       GOTO 2
CC MAKE A PRIMAL ITERATION
71     CALL RPMGAMMA(IM,JMIN)
       CALL ROWTRANS(IM,JMIN)
       GOTO 2
CC MAKE EITER A PRIMAL- OR A DUAL-TYPE ITERATION DEPENDING ON THE PIVOT
72     CALL RPMGAMMA(IM,JMIN)
       CALL RPMPHI(IMAX,JM)
       IF(IM.EQ.0 .AND. JMIN.EQ.0 .AND. IMAX.EQ.0 .AND.
      * JM.EQ.0) WRITE(IP1,200)
200    FORMAT(1X,≠ PRIMAL HAS NO FEASIBLE SCLUTICN ≠,/,
      * 1X, ≠ DUAL HAS NO FEASIBLE SOLUTION≠)
       IF(GMIN.NE.10.E6) GOTO 14
       CALL ROWTRANS(IMAX,JM)
       GOTO 2
14     IF(PHIMAX.NE.-10.E6) GOTC 15
       CALL ROWTRANS(IM,JMIN)
       GOTO 2
15     ABPH=ABS(PHIMAX)
       ABGM=ABS(GMIN)
       IF(ABPH.GT.ABGM)CALL ROWTRANS(IMAX,JM)
       IF(ABPH.GT.ABGM) GOTO 2
       CALL ROWTRANS(IM,JMIN)
       GOTO 2
99     CALL LPTEST(IM,JMIN)
       STOP 1
       END
```

```
      SUBROUTINE PHASE1
CCCCC
CCCCC THE OBJECTIVE OF PHASE I IS TO FORM A SOLUTION OVER THE LAST
CCCCC IP ROWS. ONE VARIABLE IS INTRODUCED INTO THE SOLUTION FOR EACH EQU
CCCCC AT EACH ITERATION
CCCCC
CC LAST = POINTER TO THE LAST ACCESSED TECHNOLOGICAL CCEFF. OR ONE OF IT
CC INCICATORS.
CC L = COUNTER FOR COLUMNS WITH NEGATIVE CJ-ZJ
CC GMIN IS SET EQUAL TO A LARGE NUMBER FOR INITIALIZATION PURPCSES.
CC IFST IS A FLAG INDICATING IF COLUMNS WITH NEG. CJ-ZJ HAVE BEEN PROCES
CC       OR NOT.
      INCLUDE GLOBAL2
CCC
CCC CHANGE THE SIGN OF THE COEFF. IN A EQUATICN SUCH THAT THE
CCC RIGHT SIDE IS OR BECOMES NON-NEGATIVE.
CCC
CC GO TO FIRST EQUATION OF THE TOTAL SET OF CONSTRAINTS
      LAST=PFA(LCOL)
      IF(LAST)150,151
150   IF(RESN(LAST)-MP2)155,156,156
155   IF(PPRL(LAST))157,151
157   LAST=PPRL(LAST)
      GOTO 150
CC CHECK FOR NEGATIVE RIGHT SIDE
156   IF(TCHC(LAST).GE.0) GOTO 160
      I=RESN(LAST)
      LAST1=PFR(I)
      IF(LAST1)158,99
158   TCHC(LAST1)=-TCHC(LAST1)
      IF(POUL(LAST1))159,160
159   LAST1=POUL(LAST1)
      GOTO 158
160   IF(PPRL(LAST))161,151
161   LAST=PPRL(LAST)
      GOTO 156
CCC
CCC TRY TO INTRODUCE FIRST THE MOST FAVORABLE VARIABLE FOR EACH OF THE
CCC EQUATIONS. THE EQUATIONS ARE PROCESSED IN SEQUENCE.
CCC
151   DO 100 IR=1,IP
      GMIN=10.E6
      L=1
      IM=JMIN=0
      IFST=1
      LAST=PFR(1)
      IF(LAST)1611,162
1611  IF(TCHC(LAST).GE.-TO) GOTO 163
      IND(L)=ACTN(LAST)
      L=L+1
163   IF(POUL(LAST).EQ.0) GOTO 162
      LAST=POUL(LAST)
      GOTO 1611
162   IF(ACTN(LAST).EQ.LCOL)L=L-1
```

```
      IF(L.NE.1) GOTO 102
      DO 103 J=1,N
103   IND(J)=J
      L=N+1
102   LM1=L-1
CC MAKE SELECTION AS DESCRIBED THROUGH EQUATIONS (9.11) AND (9.4).
      DO 104 K=1,LM1
      JJ=IND(K)
      THMIN(JJ)=10.E+6
      LAST3=LAST=PFA(JJ)
      IF(LAST)170,171
170   IF(RESN(LAST)-MP2)175,176,176
175   IF(PPRL(LAST))177,171
177   LAST=PPRL(LAST)
      GOTO 170
CC SEARCH OVER COLUMN JJ
176   ITMP=RESN(LAST)
      IF(ICHK(ITMP).NE.0 .OR. TCHC(LAST).LE.TC) GCTC 105
CC SEARCH B VECTOR ELEMENT CORRESPONDING TO ROW ITMP
      LAST1=LAST
      THETA=0.
182   IF(PDUL(LAST1))180,181
180   LAST1=PDUL(LAST1)
      GOTO 182
181   IF(ACTN(LAST1)-LCOL)185,186
186   TMP1=TCHC(LAST1)
      TMP2=TCHC(LAST)
      THETA=TMP1/TMP2
185   IF(THETA.GE.THMIN(JJ)) GOTO 105
      THMIN(JJ)=THETA
      IMIN(JJ)=ITMP
105   IF(PPRL(LAST))190,171
190   LAST=PPRL(LAST)
      GOTO 176
CC END SEARCH OVER COLUMN JJ
171   IF(THMIN(JJ).GE.10.E+6) GOTO 104
      GAMMA=THMIN(JJ)*TCHC(LAST3)
      IF(GAMMA.GE.GMIN) GOTC 104
      GMIN=GAMMA
      JMIN=IND(K)
104   CONTINUE
      IF(JMIN.NE.0)GOTO 106
      IF(IFST.NE.1) GOTO 107
      IFST=0
      L=1
      GOTO 162
107   IM=0
      GOTO 109
106   IM=IMIN(JMIN)
109   CALL ROWTRANS(IM,JMIN)
100   CONTINUE
      RETURN
CCC
CCC ERROR STATEMENT
CCC
99    WRITE(IP1,199)ITMP
```

```
199   FORMAT(1X,≠ERROR ... ALL ELEMENTS OF ROW≠,I3,≠ EQUAL ZERO≠)
      CALL LPTEST(IMAX,JM)$CALL EXIT
      RETURN
      END



      SUBROUTINE ROWTRANS(IM,JMIN)
CCCCC
CCCCC PERFORMS THE ROW TRANSFORMATION ON LP PROBLEM
CCCCC IM,JMIN BEING THE PIVOT INDICATORS
CCCCC
      INCLUDE GLOBAL2
      ITEST=0
      IF(ICT.GE.ITST)ITEST=1
      IF(ICT.LE.MIT) GOTO 210
      WRITE(61,211)
211   FORMAT(1X,≠MAXIMUM NUMBER OF ITERATIONS EXCEEDED≠)
      CALL LPTEST(IM,JMIN)$CALL EXIT
CCC
CCC IF ONE OF THE PIVOT INDICATORS IS ZERO, THE PROBLEM IS INFEASIBLE
CCC
210   IF(IM.NE.0 .AND. JMIN.NE.0) GOTO 201
      CALL LPTEST(IM,JMIN)
      WRITE(IP1,202)
202   FORMAT(1X,≠NO FEASIBLE SOLUTION≠,/,≠ AT LEAST ONE OF THE PIVOT ≠
     *≠INDICATORS EQUALS ZERO≠)
      CALL LPTEST(IM,JMIN)$CALL EXIT
CCC
CCC SET PIVOT ELEMENT EQUAL TO DUMMY
CCC
201   DUMMY=XTCHC(IM,JMIN)
CCC
CCC COMPUTE NEW COEFF. FOR IM ROW
CCC
      LAST=PFR(IM)
      IF(LAST)227,98
225   LAST=PDUL(LAST)
      IF(LAST)227,230
227   TCHC(LAST)=TCHC(LAST)/DUMMY
      GOTO 225
CCC
CCC DETERMINE OTHER TECH. COEFF.
CCC
230   DO 204 I=1,MP1
      IF(I.EQ.IM) GOTO 204
      DUMMY=XTCHC(I,JMIN)
      IF(DUMMY.EQ.0) GOTO 204
      LAST1=PFR(I)
      LAST2=PFR(IM)
      LAST=0
      ITMP1=0
      IF(LAST1.EQ.0 .AND. I.EQ.1) GOTO 235
      IF(LAST1)231,98
231   IF(LAST2)232,98
232   ITMP1=ACTN(LAST1)
```

```
      ITMP2=ACTN(LAST2)
      IF(ITMP2-ITMP1)235,236,237
235   TMP1=-TCHC(LAST2)*DUMMY
      ITMP3=PFR(I)
      IF(ITMP3)270,271
270   IF(ITMP1.LT.ITMP2)GOTO 274
      IF(ITMP3-LAST1)273,271
271   LAST=0
      GOTO 273
274   LAST=0
      ITMP3=ACTN(ITMP3)
      IF(ITMP3.LT.ITMP2)LAST=LAST1
273   CALL INSERT(TMP1,I,ITMP2)
      IF(ITMP1.LT.ITMP2)LAST1=LAST
      IF(POUL(LAST2))245,204
245   LAST2=POUL(LAST2)
      GOTO 232
236   TCHC(LAST1)=TCHC(LAST1)-TCHC(LAST2)*DUMMY
      IF(TCHC(LAST1).NE.0) GOTO 239
      ITMP3=PFR(I)
      IF(ITMP3.EQ.LAST1)275,276
275   LAST=0
276   CALL DELETE(I,ITMP2)
      IF(LAST.NE.0)LAST1=LAST
      IF(LAST.NE.0) GOTO 240
      LAST1=PFR(I)
      IF(LAST1)257,258
257   ITMP1=ACTN(LAST1)
      GOTO 251
258   IF(I.NE.1)GOTO 98
      IF(POUL(LAST2))260,204
260   LAST2=POUL(LAST2)
      ITMP2=ACTN(LAST2)
      GOTO 235
239   LAST=LAST1
240   IF(POUL(LAST1))250,251
250   LAST1=POUL(LAST1)
251   IF(POUL(LAST2))252,204
252   LAST2=POUL(LAST2)
      GOTO 232
237   IF(POUL(LAST1))255,235
255   LAST=LAST1
      LAST1=POUL(LAST1)
      GOTO 232
204   CONTINUE
      ICHK(IM)=JMIN
      JH(JMIN)=IM
      ICT=ICT+1
CCC
CCC CHECK IF PRINTOUT DESIRED
CCC
      IF(ICT.GE.ITST)CALL LPTEST(IM,JMIN)
      RETURN
98    WRITE(IP1,88)
88    FORMAT(1X,≠ ONE ROW IN MATRIX HAS NO NONZERO ELEMENTS≠)
      CALL LPTEST(IM,JM)$CALL EXIT
```

```
      END


      SUBROUTINE RESULTS
CCCCC
CCCCC PRINTS OUT THE OPTIMAL VALUE OF THE OBJECTIVE FUNCTICN
CCCCC             OPTIMAL VALUES OF THE PRIMAL VARIAELES
CCCCC             OPTIMAL VALUES OF THE DUAL VARIABLES
CCCCC
CC Z = OBJECTIVE FUNCTION VALUE
CC X(I) = PRIMAL VARIABLES
CC W(I)=DUAL VARIABLES
CC JH(J) = POINTER FOR EACH ACTIVITY NODE, INDICATING THE LOCATION OF IT
CC         VALUE IN THE B VECTOR OF THE SIMLEX TABLEAU
CC         IF ZERO, ACTIVITY IS NON-BASIC
      INCLUDE GLOBAL2
CCC
CCC OPTIMAL VALUE OF THE OBJECTIVE FUNCTION
CCC
      Z=0.
      LAST=PFA(LCOL)
      IF(RESN(LAST).EG.1) Z=TCHC(LAST)
CCC
CCC OPTIMAL VALUE OF THE PRIMAL VARIABLES
CCC
      DO 301 I=1,N
      X(I)=0.
      ITMP=JH(I)
      IF(ITMP.EQ.0)GOTO 301
      X(I)=XTCHC(ITMP,LCOL)
301   CONTINUE
CCC
CCC OPTIMAL VALUE OF THE DUAL VARIABLES
CCC
      DO 302 J=NP1,LCM1
      ITMP=J-N
CC PLACE THE NON-ZERO VALUES IN THE VECTOR W(I)
302   W(ITMP)=0.
      LAST=PFR(1)
310   IF(LAST)315,350
315   ITMP6=ACTN(LAST)
      IF(ITMP6.EQ.LCOL) GCTC 350
      ITMP=ITMP6-N
      IF(ITMP)316,316,307
307   W(ITMP)=TCHC(LAST)
316   LAST=POUL(LAST)
CCC
CCC OUTPUT STATEMENT
CCC
      GOTO 310
350   WRITE(IP1,305)Z,(I,JH(I),X(I),I=1,N)
305   FORMAT(1X,≠VALUE OF THE OBJECTIVE FUNCTION ≠,F13.6,/,
     *1X,≠OPTIMAL VALUES OF THE PRIMAL VARIABLES≠//(2I4,F13.6))
      ITMP=LCM1-N
      WRITE(IP1,306)(J,W(J),J=1,ITMP)
```

```
306   FORMAT(/≠ OPTIMAL VALUE OF THE DUAL VARIABLES≠//(I4,F13.6))
      CALL LPTEST(IM,JMIN)
      K=1
      RETURN
      END


      SUBROUTINE RPMPHI(IMAX,JM)
CCCCCC
CCCCCC SELECTS A PIVOT FROM AMONG THE ROWS WITH NEGATIVE BJ VALUE.
CCCCCC
CC IMAX AND JM RETURN PIVOTAL INDICATORS.
CC COMPUTATIONS AND VARIABLE NAMES ARE GIVEN BY FORMULA (9.6)-(9.8)
CC IN LLEWELYN≠S BOOK
CC PHIMAX AND DELMAX(II) ARE SET TO A VERY SMALL VALUE FOR INITIALIZATIO
CC PURPOSES.
      INCLUDE GLOBAL2
      PHIMAX=-10.E+6
      JM=IMAX=0
      KM1=K-1
CCC
CCC EACH LOOP OF DO LOOP 600 INVESTIGATED ONE ROW FOR A TENTATIVE PIVOT
CCC
      DO 600 K1=1,KM1
      II=IND1(K1)
      IF(II.EQ.1)GOTO 600
      JMAX(II)=0
      DELMAX(II)=-10.E+6
CCC
CCC START INVESTIGATION FOR ONE ROW
CCC
      LAST1=PFR(1)
      LAST2=PFR(II)
      IF(LAST2)605,99
605   IF(LAST1)606,607
606   ITMP2=ACTN(LAST2)
      ITMP1=ACTN(LAST1)
      IF(ITMP2.EQ.LCOL)GOTO 650
CCC
CCC SIMPLIFIED COMPUTATIONS FOLLOW DEPENDING ON RELATIVE POSITION OF
CCC ACTIVITIES ITMP1 AND ITMP2
CCC
      IF(ITMP2-ITMP1)610,611,601
610   TMP2=TCHC(LAST2)
      IF(TMP2.GE.-TD)GOTO 601
      DELTA=0.
      GOTO 608
611   TMP1=TCHC(LAST1)
      TMP2=TCHC(LAST2)
      IF(TMP2.GE.-TD .OR. TMP1.LT.-TD) GOTO 601
      DELTA=TMP1/TMP2
      GOTO 608
CC ALL REMAINING CJ-ZJ EQUAL ZERO
607   TMP2=TCHC(LAST2)
      ITMP2=ACTN(LAST2)
```

```
      IF(ITMP2.EQ.LCOL)GOTO 650
      IF(POUL(LAST2))660,650
660   LAST2=POUL(LAST2)
      IF(TMP2.GE.-TD) GOTO 601
      DELTA=0.
608   IF(DELTA.LE.DELMAX(II)) GOTO 601
      DELMAX(II)=DELTA
      JMAX(II)=ITMP2
CCC
CCC ADVANCE THE APPROPRIATE POINTER DEPENDING ON THE RELATIVE POSITION
CCC OF ACTIVITIES ITMP1 AND ITMP2
CCC
601   IF(ITMP2-ITMP1)630,631,632
630   IF(POUL(LAST2))635,650
635   LAST2=POUL(LAST2)
      GOTO 606
631   IF(POUL(LAST2))636,650
636   LAST2=POUL(LAST2)
      LAST1=POUL(LAST1)
      GOTO 605
632   IF(LAST1)661,607
661   LAST1=POUL(LAST1)
      GOTO 605
CCC
CCC ERROR STATEMENT
CCC
99    WRITE(IP1,639)II
639   FORMAT(1X,≠ERROR... ROW≠,I3,≠ HAS ONLY ZERO VALUES ≠)
      CALL LPTEST(IMAX,JM)$CALL EXIT
CCC
CCC DETERMINE THE MAXIMUM DELTA VALUE WITH RESPECT TO OTHER ROWS
CCC
650   IF(DELMAX(II).EQ.-10.E+6)GOTO 600
      TMP2=0.
      ITMP2=ACTN(LAST2)
      IF(ITMP2.EQ.LCOL)TMP2=TCHC(LAST2)
      PHI=DELMAX(II)*TMP2
      IF(PHI.LE.PHIMAX)GOTO 600
      PHIMAX=PHI
      IMAX=II
 600  CONTINUE
      IF(IMAX.GT.0.) JM=JMAX(IMAX)
      RETURN
      END


      SUBROUTINE RPMGAMMA(IM,JMIN)
CCCCC
CCCCC SELECTS A PIVOT FROM AMONG THE COLUMNS WITH NEGATIVE CJ-ZJ
CCCCC
CC IM AND JMIN RETURN THE PIVOTAL INDICATORS
CC COMPUTATIONS AND VARIABLE NAMES ARE GIVEN BY THE FORMULA (9.3)-(9.5)
CC IN LLEWLYN≠S BOOK
CC GMIN AND THMIN(JJ) ARE SET EQUAL TO A  LARGE NUMBER FOR INITIALIZATIO
CC PURPOSES.
```

```
      INCLUDE GLOBAL2
      GMIN=10.E+6
      IM=JMIN=0
      LM1=L-1
CCC
CCC EACH LOOP IN DO LOOP 500  INVESTIGATES ONE COLUMN WITH NEGATIVE
CCC CJ-ZJ VLUE
CCC
      DO 500 L1=1,LM1
      JJ=IND(L1)
      IF(JJ.EQ.LCOL) GOTO 500
      IMIN(JJ)=0
      THMIN(JJ)=10.E6
CCC
CCC START INVESTIGATING ONE COLUMN FOR A TENTATIVE PIVOTAL ELEMENT
CCC
      LAST1=PFA(LCOL)
      LAST2=PFA(JJ)
      IF(LAST2)505,99
505   IF(LAST1)569,507
569   IF(RESN(LAST2).GT.1) GOTO 506
      IF(PPRL(LAST2))570,99
570   LAST2=PPRL(LAST2)
506   ITMP2=RESN(LAST2)
      ITMP1=RESN(LAST1)
      IF(ITMP2-ITMP1)510,511,501
510   TMP2=TCHC(LAST2)
      IF(TMP2.LE.TO)GOTO 501
      THETA=0.
      GOTO 508
511   TMP1=TCHC(LAST1)
      TMP2=TCHC(LAST2)
      IF(TMP2.LE.TO .OR. TMP1.LT.-TO) GOTO 501
      THETA=TMP1/TMP2
      GOTO 508
507   TMP2=TCHC(LAST2)
      ITMP2=RESN(LAST2)
      IF(PPRL(LAST2))560,550
560   LAST2=PPRL(LAST2)
      IF(TMP2.LE.TO)GOTO 501
      THETA=0.
508   IF(THETA.GE.THMIN(JJ)) GOTO 501
      THMIN(JJ)=THETA
      IMIN(JJ)=ITMP2
CCC
CCC ADVANCE THE APPROPRIATE POINTER, DEPENDING ON THE RELATIVE POSITION
CCC ROWS ITMP1 AND ITMP2.
CCC
501   IF(ITMP2-ITMP1)530,531,532
530   IF(PPRL(LAST2))535,550
535   LAST2=PPRL(LAST2)
      GOTO 506
531   IF(PPRL(LAST2))536,550
536   LAST2=PPRL(LAST2)
      LAST1=PPRL(LAST1)
      GOTO 505
```

```
532    IF(LAST1)561,507
561    LAST1=PPRL(LAST1)
       GOTO 505
CCC
CCC ERROR STATEMENT
CCC
99     WRITE(IP1,539)JJ
539    FORMAT(//,1X,≠ERROR ... COLUMN ≠,I3,≠ HAS ONLY ZERO VALUES≠)
       CALL LPTEST(IM,JMIN)$CALL EXIT
CCC
CCC DETERMINE MINIMUM THETA VALUE WITH RESPECT TO OTHER COLUMNS
CCC
550    IF(THMIN(JJ).EQ.10.E+6)GOTO 500
       TMP2=0.
       ITMP2=PFA(JJ)
       IF(ITMP2.NE.0)TMP2=TCHC(ITMP2)
       GAMMA=THMIN(JJ)*TMP2
       IF(GAMMA.GE.GMIN)GOTO 500
       GMIN=GAMMA
       JMIN=JJ
500    CONTINUE
       IF(JMIN.GT.0)IM=IMIN(JMIN)
       RETURN
       END
       FUNCTION XTCHC(IM,JMIN)
CCCCC
CCCCC FETCHES A TECHNOLOGICAL COEFFICIENT FROM LINKED LIST
CCCCC
CC   IM = INDEX OF RESOURCE NODE
CC JMIN = INDEX OF ACTIVITY NODE
       INCLUDE GLOBAL2
CCC
CCC SELECT MOST LIKELY SHORTEST PATH
CCC
       IF(JMIN.GE.IM)1,20
CCC
CCC SEARCH COLUMNWISE
CCC
1      LAST=PFA(JMIN)
       IF(LAST)7,10
5      LAST=PPRL(LAST)
       IF(LAST)7,10
7      ITMP1=RESN(LAST)-IM
       IF(ITMP1)5,12,10
CCC
CCC SEARCH ROWWISE
CCC
20     LAST=PFR(IM)
       IF(LAST)27,10
25     LAST=PDUL(LAST)
       IF(LAST)27,10
27     ITMP1=ACTN(LAST)-JMIN
       IF(ITMP1)25,12,10
CCC
CCC TECH. COEFF. EQUALS ZERO
CCC
```

```
10     XTCHC=0
       RETURN
CCC
CCC TECH. COEFF. IS DIFFERENT FROM ZERO
CCC
12     XTCHC=TCHC(LAST)
       RETURN
       END



       SUBROUTINE INSERT(A,I,J)
CCCCC
CCCCC USED IN CASE A ZERO COEFFICIENT TURNS INTO A NON-ZERO ONE.
CCCCC
CCC A = TECHNOLOGICAL COEFFICIENT
CCC I = RESOURCE NODE NUMBER
CCC J = ACTIVITY NODE NUMBER
CCC LAST = POINTER TO LAST ACTIVITY NODE USED
CCC PPR = POINTER PREVIOUS RESOURCE NODE
       INTEGER PPR
       INCLUDE GLOBAL2
       ITMP1=PPRL(IAV)
CCC
CCC UPDATE PRIMAL LIST
CCC
       PPR=PFA(J)
       IF(PPR)11,12
11     IF(RESN(PPR)-I)14,15,12
CC FIRST
12     PFA(J)=IAV
       PPRL(IAV)=PPR
       GOTO 30
CC NOT FIRST
14     ITMP2=PPR
       PPR=PPRL(PPR)
       IF(PPR)20,27
20     IF(RESN(PPR)-I)14,15,27
27     PPRL(ITMP2)=IAV
       PPRL(IAV)=PPR
CCC
CCC UPDATE DUAL LIST
CCC

30     IF(LAST.EQ.0) GOTO 40
       PDUL(IAV)=PDUL(LAST)
       PDUL(LAST)=IAV
       GOTO 41
40     PDUL(IAV)=PFR(I)
       PFR(I)=IAV
CCC
CCC END
CCC
41     TCHC(IAV)=A
       RESN(IAV)=I
       ACTN(IAV)=J
```

```
      LAST=IAV
      IAV=ITMP1
      RETURN
CCC
CCC ERROR STATEMENT
CCC
15    WRITE(IP1,16)RESN(PPR), J
16    FORMAT(1X,≠A COEFFICIENT WITH COORDINATES (≠,I2,≠,≠,I2,≠) ALREADY≠
     *  ≠ EXISTS IN THE PACKED LIST≠)
      CALL LPTEST(IM,JMIN)$CALL EXIT
      END
      SUBROUTINE DELETE(I,J)
CCCCC
CCCCC USED IN CASE A NON-ZERO COEFFICIENT TURNS INTO A ZERO VALUE.
CCCCC
CC SAME VARIABLE NOTATION AS IN INSERT SUBROUTINE
      INTEGER PPR
      INCLUDE GLOBAL2
CCC
CCC UPDATE PRIMAL LIST
CCC
      PPR=PFA(J)
      IF(PPR)14,17
14    IF(RESN(PPR)-I)15,16,17
CC YES, FIRST
16    PFA(J)=PPRL(PPR)
      GOTO 30
CC NO, NOT FIRST
15    IF(PPRL(PPR))20,21
20    ITMP2=PPR
      PPR=PPRL(PPR)
      IF(RESN(PPR)-I)15,26,17
26    PPRL(ITMP2)=PPRL(PPR)
      GOTO 30
CC LAST COEFF. REACHED
21    IF(RESN(PPR)-I)17,22,17
22    PPRL(ITMP2)=0
CC CHANGE POINTER COMMON TO ALL THREE CASES ABOVE
30    PPRL(PPR)=IAV
      IAV=PPR
CCC
CCC UPDATE DUAL LIST
CCC
      IF(LAST.EQ.0) GOTO 40
      PDUL(LAST)=PDUL(IAV)
      GOTO 50
40    PFR(I)=PDUL(IAV)
CCC
CCC END
CCC
50    RESN(IAV)=ACTN(IAV)=PDUL(IAV)=0
      TCHC(IAV)=0.
      RETURN
CCC
CCC ERRORO STATEMENT
CCC
```

```
17     WRITE(IP1,18)
18     FORMAT(1X,≠PACKED LIST DOES NOT CONTAIN THE NON-ZERO ≠
      *  ≠COEFFICIENT SEARCHED≠)
       CALL LPTEST(IM,JMIN)$CALL EXIT
       END
       SUBROUTINE WRPACK
CCCCC
CCCCC WRITES OUT THE LIST STRUCTURE FOR THE PACKED LF MATRIX
CCCCC
CC VARIABLE DESCRIPTION SAME AS FOR PACK ROUTINE
       INCLUDE GLOBAL2
       WRITE(IP1,59)IAV,LAST,LAST1,LAST2
59     FORMAT(//1X,≠IAV =≠,I5,≠ LAST =≠,I5,≠ LAST1 =≠
      *,I5,≠ LAST2 =≠,I5)
       WRITE(IP1,57)M,IP,N,LCOL,ITST,MIT
57     FORMAT(6I5//)
       WRITE(IP1,55)(J,RESN(J),ACTN(J),TCHC(J),PFRL(J),POUL(J),J=1,NL)
C  WRITE POINTERS OUTSIDE PACKED MATRIX
       WRITE(IP1,56)(I,PFR(I),I=1,MP1)
       WRITE(IP1,56)(J,PFA(J),J=1,LCOL)
55     FORMAT(3I5,F10.3,2I5)
56     FORMAT(1X,12(I7,I4))
       RETURN
       END
       SUBROUTINE LPTEST(IM,JMIN)
CCCCC
CCCCC UNPACKS THE LINKED LIST AND PRINTS IT OUT IN
CCCCC TABLEAU FORMAT
CCCCC
       INTEGER PRES
       INCLUDE GLOBAL2
       WRITE(IP1,10)ICT
       WRITE(IP1,15)
15     FORMAT(//)
10     FORMAT(////1X,≠ITERATION NO. ≠,I3/)
CCC
CCC START COMPOSING MATRIX ROW BY ROW
CCC
       DO 1 L=1,MP1
       PRES=PFR(L)
CC ZERO BUFFER
       DO 2 K=1,LCOL
2      BUF(K)=0
CC PLACE NON-ZERO COEFF INTO BUFFER
4      IF(PRES.EQ.0) GOTO 3
       ITMPJ=ACTN(PRES)
       BUF(ITMPJ)=TCHC(PRES)
       PRES=POUL(PRES)
       IF(PRES.LE.NL) GOTO 4
       WRITE(IP1,7)
7      FORMAT(1X,≠UNPACK ROUTINE GOES BEYOND LIMITED FREE ≠
      *,≠STORAGE SPACE≠)
       CALL EXIT
CCC
CCC WRITE BUFFER OUT
CCC
```

```
3       WRITE(IP1,12)(BUF(J),J=1,LCOL)
        WRITE(IP1,17)
17      FORMAT(/)
1       CONTINUE
CC WRITE OUT MOST IMPORTANT ITERATION VALUES
        WRITE(IP1,13)IM,JMIN,
       *GMIN,PHIMAX,THETA,GAMMA,DELTA,PHI,TO
        WRITE(IP1,14)(ICHK(I),I=2,MP1)
12      FORMAT(1X,16F8.2)
13      FORMAT(1X,≠PIVOT ELEMENT≠,2I4,/1X,
       *≠GMIN ≠,E11.3,≠   PHIMAX ≠,E11.3,≠   THETA ≠,E11.3,/,
       *≠   GAMMA ≠,E11.3,≠   DELTA ≠,E11.3,≠   PHI ≠,E11.3,
       *≠   TO ≠,E7.3)
14      FORMAT(1X,≠ICHK≠,10I4)
        RETURN
        END
```