

2007

# Database Structure for a Class of Multi-Period Mathematical Programming Models

Goutam Dutta

*Indian Institute of Management, Ahmedabad*

Robert Fourer

*Northwestern University, Evanston*

Follow this and additional works at: <http://aisel.aisnet.org/icdss2007>

---

## Recommended Citation

Dutta, Goutam and Fourer, Robert, "Database Structure for a Class of Multi-Period Mathematical Programming Models" (2007).  
*ICDSS 2007 Proceedings*. 9.

<http://aisel.aisnet.org/icdss2007/9>

This material is brought to you by the International Conference on Decision Support Systems at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICDSS 2007 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

**Title:** Database Structure for a Class of Multi-Period Mathematical Programming Models

**Author 1:** Goutam Dutta

Indian Institute of Management, Ahmedabad

Vastrapur, Ahmedabad 380015, India

**Author 2:** Robert Fourer

Department of Industrial Engineering and Management Science

Northwestern University, Evanston, IL 60208, USA

### **Abstract**

We describe how a generic multi-period optimization-based decision support system can be used for strategic and operational planning in a company whose processes can be described in terms of five fundamental elements: Materials, Facilities, Activities, Times and Storage-Areas. We discuss the issues of interface design, data reporting and updating, and production and profit planning. We also compare the performances of two different types of database structures with respect to optimization.

**Keywords:** Decision Support System, process industries, optimization, strategic and operational planning.

### **1. Introduction**

This work started as a project to design an optimization-based decision support system (DSS) for strategic planning for steel companies in North America. As the project was supported by The AISI (American Iron and Steel Institute), the DSS was generic in concept, but capable of being specifically applied to any particular company's facilities by supplying appropriate data Fourer [9]. Complete data for a steelmaking operation, including values such as yields, capacities, and prices indexed over products and processes could be conveniently supplied in the form of relational database files. The optimization was taken to be over a

single planning period, however, and thus difficulties involving the indexing of data and model entities over time were not addressed.

This paper extends the work of Fourer [9] to a multi-period case. We adopt the fundamental elements of Materials, Facilities, and Activities from the previous work, but add two more elements - Times and Storage-Areas. Among the major points we address are the following:

- What are the key features of a multi-period DSS?
- What are the difficulties in implementing a multi-period DSS?
- What are the alternatives for handling multi-period indexing in a database context?
- In what ways can the optimal result be represented in a multi-period DSS?
- Why is an update mode difficult in a multi-period database?
- How do alternative data structures compare with respect to data storage, data retrieval, and support for optimization models?

## **1.1 Literature Review**

Database representation of an LP is one of the eight approaches of LP representation comprehended by Murphy [12]. The approach is considered as translation form, as it is used as a bridge between modelers' form, and algorithmic form. Fourer [8] recognized that no single form of LP representation can be developed which can be easily understood by modelers', computers, and industry practitioners simultaneously. Database representation of LP was initially discussed by Fourer [8]. Murphy [12] summarized eight most popular approaches of representing an LP. He summarized the modelers' form, algorithmic form, and translation forms of LP representation.

Most of the industry data remains in database systems, one need to look for a system which can handle the bulk data required for an LP in a systematic manner. Data modeling in context with mathematical programming is discussed in detail (Dominguez, Mitra, and Lucas [5]). Readers interested in database systems, and data modeling are referred to the book by Date [2]. We recognize that not much work is done

in representing an LP in translation form in general and database form in particular. We demonstrate (using an LP model for strategic planning in process industry), how an LP can be represented in the form of a database structure. We follow the widely accepted Dolk [3, 4] framework for data, model, and dialogue management.

## **1.2 Outline**

In section 2 of this paper, we discuss the design issues raised by a multi-period database. We introduce the various elements of the DSS and discuss possible implementations. We also consider the correspondences of the various files in the DSS with the various variables in the linear program. In section 3, we discuss the various steps of multi-period optimization - constraint and variable generation, coefficient matrix generation, solution of the optimization problem, and reading of the optimal values back into the database. We also indicate how we allow for soft capacities through the use of artificial variables. Section 4 considers how the various features of the DSS can be useful for the strategic and operational planning in a process industry. Section 5 discusses the various features for reporting and updating of the data, and in section 6, we compare two different variations of the database design – one primarily hierarchical and another primarily relational - with respect to optimization. We conclude the paper by outlining the scope for further work.

## **2. Database Design Issues for Multi-Period Models**

Our generic multi-period planning model has, as previously noted, five fundamental elements:

*Times* are the periods of the planning horizon, represented by discrete numbers (1, 2, 3 ...). They can be as short as weeks, though for a planning model they are most likely to range from months to years.

*Materials* are the physical items that figure in some stage of production. They may be inputs, intermediates, or outputs, and sometimes more than one of these.

*Facilities* are collections of machines that produce some materials from others. For example, a Hot Mill that produces sheets from slabs is a facility.

*Activities* are productive transformations of materials. Each facility houses one or more activities, which uses and produces materials in certain proportions. Production of hot metal, production of billets, pickling, and galvanizing are examples of steelmaking activities.

*Storage-Areas* are fields or warehouses where raw materials, intermediate products, or finished products may be stored.

An earlier paper Fourer [9] describes the algebraic formulation and corresponding database structure for the single-period version of this model, which has to deal with only materials, facilities, and activities. An algebraic formulation for the multi-period model is given in **Appendix**.

In this section we describe the database changes and extensions that have to be made to accommodate the times and storage-areas. The *Times* in particular are a different kind of entity whose addition poses a number of difficulties. Some alternatives to the structure given here will be discussed in Section 6.

## **2.1 The *Times* file**

Our database is implemented within 4th Dimension, a relational database management system, Adams [1]. Other database systems such as Access or Oracle could be used just as well. Figure 1 summarizes the structure of the database as expressed within 4th Dimension. The five boxes labeled *Materials*, *Facilities*, *Activities*, *Times*, and *Storage-Areas* correspond to the five major Elements, or files, of the database. Items within each box denote the file's data fields and subfiles, with the subfile entries distinguished by a light-shaded line that runs to the top of a separate box in which the subfile data fields are listed. The smaller, independent database structure in the upper right of the diagram holds a generated linear program as described at the end of this section.

Following 4th Dimension's notation, we use bracketed names to denote files and apostrophes to separate subfile and field names. Thus [Facilities] is the database file of facilities, [Facilities]Inputs is the subfile of

[Facilities] file, and [Facilities]Inputs'InMin is a data field of the subfile. The presence of subfiles implies a partially hierarchical rather than purely relational structure to the database; further details can be found in the earlier discussion of the one-period model Fourer [9].

The structure of the Times file in the database (Figure 2) is very simple, consisting basically of a record per period. A name field can be adjusted according to whether the periods are modeling, say, quarters or years. The complications introduced by the multi-period structure lie mainly in the ways that Times interact with all of the other pieces of the database structure.

**Insert Figure 1 and 2**

## **2.2 Materials File**

The Materials data (Figures 3 and 4) are stored in a hierarchical way. In Table 1, we show the one-to-one correspondence between the parameters of the LP model and the fields in the [Materials] file. In this file the material name ([Materials]MatName) and material identification string ([Materials]MatTag) are unique. [Materials]MatTag is required for data entry in the sub-files of the [Materials] file. In the [Materials] file, BuyMax, BuyMin, SellMax, SellMin, BuyPrice, SellPrice, BuyOpt, SellOpt, InvMax, InvMin, InvOpt, InvCCost, CostIn and CostOut are the time-dependent subfields in the [Materials]MatTime sub-file. Since the model is multi-period, the dual variables (MatDual) are also considered as a function of time and are put in the MatTime sub-file. MatName, MatUnits and MatType are the main fields of the file. MatTimeID is the indexed subfield of the Materials[MatTime] sub-file. For each material, there is a record of the [Materials]MatTime sub-file corresponding to each record of the Time file; the data in each [Material]MatTime'MattimeID sub-file field is the same as the value in the corresponding [Time]TimeID field.

**Insert Table 1**

**Insert Figure 3 and 4**

The sub-file Conversions is the second sub-file of [Materials] file. This sub-file is indexed by two subfields: Conversion time (ConvTime) and Conversion Material (ConvTo). In addition, it has conversion cost

(ConvCost) and conversion yield (ConvYield) as additional subfields. The [Materials]Conversions sub-file is similar to the analogous sub-file in the single period model in STEEL (Figure 5) except that it has the additional subfield [Materials]Conversions'ConvTime and is indexed over Times as well as Materials.

The Materials file has a third sub-file called [Materials]Compositions. In this sub-file, we have [Materials]Compositions.'CompName and [Materials]Compositions'CompTime. [Materials]Compositions'CompName is the time-dependent subfield of the subfile. The maximum and minimum compositions of each element or compound are the two additional subfields. These subfields are required for the *Cost Allocation Model* that we do not discuss in this paper.

**Insert Figure 5**

### **2.3 Facilities File**

In the Facilities file (Figures 6 and 7), for time dependent parameters we retain a structure similar to that of the [Materials] file. In Table 2 of Appendices, we show the one to one correspondence between the parameters of the LP model and the fields [Facilities] file. We define [Facilities]FacTime as a subfile where the CapMax, CapMin, CapOPT and CapDual subfields are the time dependent maximum, minimum, and optimal production levels of the facility, and the time dependent dual value of the facility capacity. The VendorCost is the cost of vendoring (outsourcing) an additional unit capacity of the facility at that time.

There are two indexed subfields in [Facility]Inputs, which is a sub-file of the Facilities file. The first one is the input material, which is related to the [Materials] file. The other is [Facility]Inputs'InTime which is the time dependent field of the [Facilities]Input File and is related to the Time file. The subfile [Facilities]Outputs is entirely analogous.

**Insert Figure 6 and 7**

**Insert Table 2**

## 2.4 Activities File

[Activities] is defined (Figure 8 and 9) as a separate file. (In STEEL-TIME2, we consider [Activities] as a sub-file of the [Facilities] File). There is a field of [Activities]ActTime which is the indexed field of time in the [Activities] file and related to the [Times] file. In Table 3 we show one to one correspondence between the parameters of the LP model and the [Activities] file. In each [activities] file there is a field; ActFacName that specifies which facility it belongs to. This is required so that the user can search for the activity through the facility. The other important field is ActTag, the unique identification of each activity. The [Activities] file can be indexed over [Activities]Act Name or [Activities]ActTag (identification string). Two activities may have the same ActName (like PRODUCTION OF BILLET), but if they have a different ActTime, they will have a different ActTag. In other words every record of [Activities] file will be identified by a unique [Activities]ActTag.

While defining the activity inputs (ActInMat) or activity outputs (ActOutMat), we have to consider the fact that ActInMat (or ActOutMat) should have only those materials which are in Facility Inputs (or Outputs) and also at the time where ActTime is equal to [Facilities]Inputs'IntTime ([Facilities]Outputs'OutTime). For example, let us assume that the BLOOM, BILLET and SLAB are available as [Facilities]Inputs at Time =1 in [Facilities]FacName =ROLLING MILL, but BLOOM and SLAB are only available as [Facilities]Inputs at Time =2 in the same facility. In the [Activities]File at Time=1 the possible choices available in the subfield [Activities]ActInPuts'ActInMat are BLOOM, BILLET and SLAB, but only SLAB and BLOOM are available as [Activities]ActInputs'ActInMat at Time =2 in the same facility.

**Insert Table 3**

**Insert Figure 8 and 9**

## 2.5 Storage-Areas File

In the [Storage-Areas] file (Figures 10 and 11) we have the name of the Storage-Area and the time at which the materials are stored. In addition, we have the capacity constraint of the storages giving the maximum and the minimum capacities of the storage-areas. The structure of the [Storage-Areas] file is similar to that of the



[Activities]File. [Storage\_Areas]StoreTag is the field which uniquely identifies the records of the file. In the [Storage-Areas] file, we have a sub-file called [Storage-Areas]StoreMatList which lists all the materials that can be listed. In Table 4 we show one to one correspondence between the parameters of the LP model and the [Storage\_Areas] file.

**Insert Figure 10 and 11**

**Insert Table 4**

## **2.6 Variables File**

In the [Variables] file (Figure 12) we have fields Number, Type (Material Bought, Material Sold, Material Inventoried, Activity at Facility), Identification Number 1 (ID1), Identification Number 2 (ID2), Objective, Upper bound and Lower bound as in the single period model. However, we have also an Identification Number 3 (ID3) field which indicates the time of the variable. [Variables]Optimal refers to the most recent optimal value of the variable. The variables file has a sub-file known as [Variables]Coeffs which has a subfield called [Variables]Coeffs'Constr and this constraint is related to the [Constraints]Number of the [Constraints] file.

**Insert Figure 12**

## **2.8 Constraints File**

In addition to [Constraints]Number, the [Constraints] file (Figure 13) has a field for Type (Material Balance, Facility Input, Facility Output and Facility Capacity, Storage Capacity or Storage Total, which refer to the equation numbers 3-8 respectively in Appendix). The Identification Number 1 ([Constraints]ID1) indicates the Material Name for the Material Balance equation and Facility Name for the other three types of constraints. The Identification Number 2 ([Constraints ID2) refers to the material for the Facility Input and Output respectively. As in the [Variables] file, ID3 refers to the time of the variable. [Constraints]Dual refers to the dual variable corresponding to the most recent optimal solution.

**Insert Figure 13**

### 3. Optimization

Once the data of the five database files and their respective sub-files are entered, they are validated by a set of diagnostic tests to be explained in the next sub-section.

#### 3.1 Optimization Steps

This subsection describes how the subsequent optimization process is carried out. The principal steps (Figure 14) are as follows:

##### *Insert Figure 14*

1. The data describing the production scenario at different time periods is collected and stored in the database.
2. The constraints associated with the linear program are generated. The constant terms of the constraint equations or inequalities, LoRHS and HiRHS, are extracted from the database and stored in the [Constraints] file.
3. The variables of the associated linear program are determined, along with their coefficients in the constraints. Variables are stored in a separate [Variables] file and coefficients in its [Variables]Coeff subfile. This step gives the user a choice of discounted or undiscounted optimization. If the latter is chosen, it prompts for an interest rate, and all cost, price, and revenue data are converted to their discounted values in the objective function.
4. The [Constraints] and [Variables] files are scanned and all of the essential information about the linear program is written to an ordinary text file in a compact format. This text file is the input file to our solver.
5. A linear programming solver reads the text file - we used XMP, by Martsen [11] - which solves the indicated linear program and then writes the optimal values of the variables to a second text file.
6. The second text file is read and the optimal values are placed in appropriate fields of the [Materials], [Facilities], [Activities], and [Storage\_Areas] files and their sub-files.

To support these activities, the database offers three modes of display. The Data mode is primarily for entering data describing the operations to be modeled. The Optimal mode shows the fields for the optimal values, and hence is intended for examination of results. Finally, an Update mode allows small changes to be made to the data without a time-consuming re-generation of the [Constraints] and [Variables] files.

### 3.2 Diagnostics Rule

The diagnostic routines are written to ensure that the linear program is complete and free from errors and infeasibilities. We use the various file procedures, layout procedures and global procedures to implement these routines.

The following generic diagnostics are applied to all files and sub-files or variables or constraints, as appropriate:

**Rule 1:** For every variable the upper bound should not be less than the lower bound. For every constraint the lower right hand side (LoRHS) should not be more than the higher right hand side (HiRHS).

**Rule 2:** For every variable and every constraint, there should not be more than one non-zero element.

**Rule 3:** For every sub-file indexed over one time subfield, the number of sub-records in the sub-file should be same as the number of records in the [Times] file.

**Rule 4:** For files and sub-files indexed over one time field and one non time field, the number of records (or sub-records) should not be more than the product of the number of records (or sub-records) in the [Times] file and the number of records related to the non-time field.

**Rule 5:** If a record or sub-record is indexed over a time field or sub-field and one non-time field or sub-field, there will be only one record or sub-record containing any particular combination of the time field and non-time field.

We assume that the linear program is complete with respect to all time period data. If we do not have data for any period, a default value is taken. The default values of all minimums are zero and of all maximums are infinity (implemented as 99999999). The default value of yield is 100 % and of rolling rate is 1 ton/ hour.

## **4. Features of the DSS**

We would like to use this DSS for strategic and operational planning. In this subsection, we will discuss various features of this DSS.

### **4.1 Strategic and Operational Planning**

In strategic planning, the DSS will be able to answer questions such as:

1. What is the effect of cost or price changes of raw materials and finished products on the product-mix?
2. If we invest 20 million dollars to install a coal injection system in the blast furnace this year, anticipating an increase of productivity of the blast furnace by 5 percent in subsequent years, is the investment justified?
3. If the company is planning to diversify into different products, what products should be chosen?

In operational planning the DSS will be able to help the steel company officials with questions like these:

1. How does product-mix planning for the current month affect planning in the subsequent months and can this monthly plan be divided into four weekly plans or even daily plans for 30 days?
2. In response to a shortage of liquid steel, which results in the partial operation of the finishing mills in the downstream production line, which of the finishing mills should go down?
3. Should external scrap be purchased as a substitute for hot metal and at what price?

For example, in the experience with an Indian steel company (Sinha [14], Dutta [7]) the marginal profit of an extra megawatt of electrical power was found to be several million dollars. This study justified the investment of installing diesel-generating sets. Similar studies can be done using our DSS.

## 4.2 Soft Capacities

If we have infeasibility in the "Facility Capacity" constraint, we can generate a "Soft Capacity" variable, which is similar to an artificial variable. At the end of step 2, the user will have the option to use a procedure which generates this variable.

This procedure will generate  $X_{i,t}^{vend}$  (the soft capacity variable) in the Facility Capacity (Constraint 6 of Appendix) and will also generate its related objective function coefficients. The user needs to enter the value of [Facilities]FacTimeVendorCost which is the coefficient of the soft capacity variable in the objective functions. In case we do not want the capacity constraint to be violated, we assign a very high value to these objective function coefficients.

## 4.3 User Friendliness

This is the most important point of this research. We have been able to demonstrate that multi-period, multi-product, multi-facility process industry planning can be done with little or no knowledge of linear programming. All the user has to do is click the appropriate buttons to run the required linear programs.

The DSS can be used in three modes: *Data*, *Optimal* and *Update*. In the *Data* mode, the user enters data in the five different files. The *Optimal* mode is for display of optimal values and dual prices. The DSS takes much longer (92 minutes) to generate the [Variables] file and the [Constraints] file than to solve the problem (3 minutes). If there is no addition or deletion of records in the [Materials], [Facilities] and [Activities] file, any change in the parameters of these files can be reflected in the corresponding changes to the [Variables] and [Constraints] file (without procedures of variable and constraint generation). This is accomplished in the *Update* mode resulting in saving of user time.

As a user-friendly tool for strategic planners, the dual prices for "Facility Capacity" constraints for each facility are displayed to indicate the profit improvement potential. The details of the dual prices are explained in sub section 5.3.

#### **4.4 Multi-Period Model**

The multi-period structure of our DSS has the following advantages:

1. The model can show how the cash flow of the company changes with different interest rates. The user is allowed to enter the interest rate. The user also has the option to optimize over nominal or discounted financial parameters.
2. The importance of inventories is considered in this model. Using this DSS we will be able to make decisions as to whether it is more profitable to produce at the current time period and hold inventory, or to produce in the future.
3. The user can see the effect of changing the parameters in one time period on the optimal decisions for other time periods.

#### **4.5 Generality and Flexibility**

The model is sufficiently generic so that it can be used by any process industry that transforms materials in different facilities. When the company decides to make any new product, a record can be added to their materials database. Similarly when a new facility is installed the user can enter an appropriate record. For any linear programming model done in AMPL or GAMS or Excel Solver the user does not have the advantage of route flexibility. In this DSS, any route of the product can be added or deleted by addition and deletion of appropriate material, facility and activity. If another industry wants to use this software, they only need to change the relevant data entry files for their company.

### **5. Reporting and Updating the Data**

In this section, we consider the different files and discuss the time dependent layouts where the time dependent parameters are entered as subfields.

#### **5.1 Layouts with Time as a Subfield**

First, let us consider the [Materials] File. In this file, no time dependent parameters are in the file level except for MatInvZero. This field is required to initialize the linear programming model.

[Materials]MatTime is a sub-file which is indexed over time, so we have designed a layout that displays all the time-dependent parameters that are in the subfields in this sub-file (Figure 15). These fields will be the same in *Data* or *Optimal* layouts. In order to see the optimal value of the material COIL bought at Time = 2, the user has to select the optimal mode in the *Examine* menu of the main menu and select Materials. Then a list of Materials will be displayed. The user has to then select the material COIL and a layout called Materials Optimal (Figure 16) will be displayed. In this layout there will be an included layout that lists the data of all time dependent parameters of the materials COIL. Once the user selects Time=2 a list of parameters is displayed in a layout for Time=2 and one of them is BuyOPT which shows the optimal value of Material bought in Time= 2. Similarly, if the user wants to get the BuyPrice of material called SCRAP at Time =3, he or she has to go through steps similar to all these.

We now discuss two different types of searches. We want to compare the searching process of an activity and an input material in the same [Facilities] file. Let us assume that [Facilities]FacName= BASIC OXYGEN FURNACE. The user selects Facilities and Optimal in the *Examine* menu of the main menu and gets a listing of all facilities and selects the facility = BASIC OXYGEN FURNACE and goes to the Facilities Optimal screen.

#### **Insert Figure 15 and 16**

This is common to both the searches. In the first search, he or she clicks the Activities button and goes to the next page of the Facilities Optimal Screen. This screen layout lists all the activities in this facility as an included layout. If the user wants to find the values of rate for the output material STEEL for the Activity = CRUDE STEEL PRODUCTION at Time=2 of this facility, then he or she looks at the list of activities and searches for Activity = CRUDE STEEL PRODUCTION and Time=2. This leads to an Activity Optimal Screen which lists the output materials. Then this list gives the value of output rate for the output material =STEEL. *In this case, to get a required value, we first search (on the [Activities] file) with a combination of two fields, and then look for a sub-file or subfield.* In the second search, to get the maximum value of input material STEEL SCRAP that can be accommodated in this facility at Time=2, the

user looks at the facilities Optimal Screen and looks at the included layout of Inputs. This included layout lists all the input materials at all times. The user then searches for Material = STEEL SCRAP and Time=2. *In this case the search is performed with two searches at the sub-file level.*

## **5.2 Included Layouts and Graphs in the Time File**

Suppose we have a question from a user. At Time =1, what is the optimal value of material sold for SINTER, and HIGH CARBON BILLET? In the Examine menu, the user can select Materials and *Optimal*, and this will lead to a list of Materials. The User can double click at SINTER and this will lead to the Materials Optimal screen of SINTER. In this screen there will be a list of Times and the user can find the optimal value of material sold at Time = 1 in this list. Then he has to return to the list of Materials and double click here again at HIGH CARBON BILLET. Then he gets another Materials Optimal Screen of HIGH CARBON BILLET. Then he can look again at the Time Layout and see the material bought at Time = 1. This is a cumbersome procedure. At Time=1, the user cannot go from one material to another. This can be overcome by making an included layout of the [Materials] file in the [Times] file.

In the 4th Dimension database management system, we have the advantages of using an included layout. In an included layout, the layout of one file can be included in another file. So we can see the [Materials] file or the [Facilities] file as an included layout in the [Times] File. In this case, the user selects Time-Material at the Examine menu. This leads to a list of times. The user selects Time=1 and he or she is supplied with a list of [Materials] at Time=1 (Figure 17 and 18). In this case the user can switch from one material to another at the same time (Time=1).

### **Insert Figure 17 and 18**

Similar arrangements can be made for the [Facilities] file and similar advantages can be achieved by making the [Facilities] file as an included layout of the [Times] File (Figures 19 and 20).

### **Insert Figure 19 and 20**

While discussing optimal layouts, we also consider the case of graphs. We can display the graphs of different variables, such as materials bought, and materials sold. We have tried two different types of graphs, the line



graph and the bar graph. Similarly, we can display graphs of the material inventory (Figure 21). Other than these, we can display the maximum, minimum, and the optimal values from the [Facilities] Inputs or [Facilities]Outputs sub-file of the [Facilities] file. The graphs give the user an idea about where the optimal value lies and how close the optimal value is to the maximum.

**Insert Figure 21**

### 5.3 Reporting of Optimal Dual values

In this section, we discuss the difficulties in reporting the optimal dual values in multiple time periods. For a single period model, the display of dual values is simple and straightforward. However, for the multi-period model we have dual values for more than one time period. In addition, the reduced cost for the variable Material Inventoried, any time period is dependent on dual values from more than one time period. This makes our task difficult for displaying the optimal dual values.

Let the reduced costs for the Material Bought, Material Sold and Material Inventoried at time t be denoted by  $RC_{jt}^{buy}$ ,  $RC_{jt}^{sell}$ ,  $RC_{jt}^{inv}$  respectively, and let  $\Pi_{jt}$  be the dual price of the material balance equations for material j at time t. Then

$$\begin{aligned}
 RC_{jt}^{buy} &= \Pi_{jt} - C_{jt}^{buy} \\
 RC_{jt}^{sell} &= C_{jt}^{sell} - \Pi_{jt} \\
 RC_{jt}^{inv} &= h_{jt} - \Pi_{jt} + \Pi_{j,t-1}
 \end{aligned}$$

Reduced costs are the dual values on the bounds. As the dual values on material balance constraints are available with the solution of the optimization problem, the reduced cost values of the variables (Material Bought, Material Sold and Material Inventoried) can be easily computed. The computation of reduced cost for material inventoried is slightly difficult as we need to store dual values for more than one time period, but we can use global procedure and scripts to overcome this.

As we have explained earlier in sections 4.1 and 4.2, we can display the dual values (Figure 17, 18, 20) and the reduced costs in a layout for [Facilities] that contain a scrolling list of times or in a layout for [Times] in a scrolling list of facilities.

#### 5.4 Optimal Summaries

In the case of a multi-period model, creation of summaries is difficult and not straightforward like in single period models. In this section, we discuss two different ways the summaries can be displayed: summaries of each time period separately, and grand summaries for all time periods.

We repeat the equation of the objective function (**equations of Appendix**):

$$\begin{aligned}
 Z(t) = & \left( \sum_{J \in M} c_{jt}^{sell} x_{jt}^{sell} - \sum_{J \in M} c_{jt}^{buy} x_{jt}^{buy} - \sum_{(j,j') \in M^{conv}} c_{jj't}^{conv} x_{jj't}^{conv} - \right. \\
 & \left. \sum_{(i,k) \in F^{act}} c_{ikt}^{act} x_{ikt}^{act} - \sum_{j \in M} h_{jt} x_{jt}^{inv} - \sum_{i \in F} C_{i,t}^{vend} x_{i,t}^{vend} \right)
 \end{aligned}$$

(1)

Z=

$$\boxed{\sum_{t \in T} Z(t)}$$

2)

We will now break it up into different parts. Typically a user would like to answer "What is the sum total of revenue obtained by selling all materials at one time (say Time = t)?" This figure can be obtained by searching for [Times]TimeID = t and summing over all the materials(is there anything missing here?) the quantity [Materials]MatTime'SellPrice multiplied by [Materials]MatTime'SellOPT. This will indicate the revenue obtained from the sale of all the materials at this time. Let us define it as it R(t), the revenue at time t :

$$R(t) = \sum_{J \in M} s_{Jt}^{sell} x_{Jt}^{sell}$$

(3)

Similarly we can write the corresponding summation terms for the other terms. We define

$Cp(t)$  = Cost of purchase of all materials at time t

$Ca(t)$  = Cost of all activities at time t

$Ci(t)$  = Cost of carrying inventory at time t

$Cc(t)$  = Cost of conversions at time t

$Cv(t)$  = Cost of outsourcing at time t

Once we have calculated all the six quantities we can rewrite the net profit as the following:

$$Z(t) = R(t) - Cp(t) - Ca(t) - Ci(t) - Cc(t) - Cv(t) \quad (4)$$

The terms of equation 5.5 can be displayed in a grand summary over all time periods (Figure 22, 23). Based on the equation 5.8, we can also display the summary for each period (Figure 24).

**Insert Figure 22, 23, and 24**

### **5.5 Discounted Cash Flow and Capital Budgeting Issues**

The advantage of the multi-period model is that we can incorporate the time value of money. In a financial analysis, if there is no time value of money, we call the results a nominal cash flow. In a discounted cash flow, the user can choose the interest rate (Figure 25). The summary statement for each time and the grand summary statement can be converted to the discounted cash flows (Figure 26) and discounted summaries (Figure 27).

With the features of the DSS, we can have any one of the three alternatives given below:

1. Optimize with the nominal objective function and display the optimal result as a nominal cash flow without considering discounting.
2. Optimize with the discounted objective function and display the optimal result in a nominal cash flow statement. In this case optimization is performed after discounting.

3. Optimize with the nominal objective and convert the optimal result to discounted cash flows and show the discounted cash flows. In this case the discounting is done after the optimization.

**Insert Figure 25, 26, and 27**

## **6. Comparison of Database Structures**

In this section, we consider the different variations of the [Materials] and [Facilities] files. These files can be organized in several ways and we discuss how the computer times for variable and constraint generation vary with different variations of the relational and hierarchical databases. We consider two different types of structures: STEEL-TIME1 and STEEL-TIME2. The structure of STEEL-TIME1 is similar to STEEL-TIME (which we have discussed in Section 3. Fourer (1997) has studied two different variations of the [Constraints] and [Variables] files, one relational and one hierarchical. We extend his comparison to two different variations of the [Materials] and [Facilities] files. We compare the implementation of STEEL-TIME1 and STEEL-TIME2 according to four different criteria: ease of use, data storage and retrieval, ease of development and efficiency of optimization.

**Insert Figure 28 and 29**

### **6.1 Implementation of STEEL-TIME1 vs. STEEL-TIME2**

STEEL-TIME1 is a modified version of STEEL-TIME. We find that STEEL-TIME2 is faster in generating the variables and constraints than STEEL-TIME1. This is because in STEEL-TIME1, the data for time dependent parameters are stored in a sub-file. So every time a record is written in the [Variables] file, first the record of the [Materials] is searched for, then the sub record of the file is searched for, and finally the record is written in the [Variables] file. However in STEEL-TIME2, fields like BuyMax, BuyMin are at the field level. Therefore to write a record in the [Variables] file, we only have to search the [Materials] and [Facilities] at the file level. Similarly, the disk-space for the data of STEEL-TIME2 is higher than that of STEEL-TIME1. This is because time-independent parameters like MatName, MatTag, MatInvZero, FacName, FacTag, FacType etc. are duplicated in STEEL-TIME2.

The numbers of constraints and variables in STEEL-TIME1 and STEEL-TIME2 are equal. The other similarities and differences of STEEL-TIME1 and STEEL-TIME2 are as follows:

1. In STEEL-TIME1, the time dependent parameters are in subfields of the [Materials] and [Facilities] files. In STEEL-TIME2 these are in the fields of the [Materials] and [Facilities] files.
2. The [Storage-Areas] file of STEEL-TIME is not considered in this comparison. In addition, vendoring or outsourcing is not considered an option. Even if the indexing in the formulation and the way of representing the mathematical model are different, we essentially solve the same optimization problem in STEEL-TIME1 and STEEL-TIME2.
3. STEEL-TIME1 or STEEL-TIME2 cannot be clearly classified as a purely relational or purely hierarchical database. Each has both relational and hierarchical aspects. STEEL-TIME1 is more relational and [Activities] is a separate file. STEEL-TIME2 is more hierarchical, and [Activities] is a sub-file of the [Facilities] file.

## **6.2 Ease of Use**

STEEL-TIME1 appears to be more complicated than STEEL-TIME2. Other than the [Times] file there are only two files in STEEL-TIME2, the [Materials] and the [Facilities] file.

Therefore it is easier to use STEEL-TIME2 than STEEL-TIME1. In the [Materials] file, all the purchase, sales and inventory related data about the Materials are kept at the file level. When the materials are displayed on an output layout, in STEEL-TIME2, sorting is possible with respect to the [Materials]MatName as well as [Materials]MatTmeID. However in STEEL-TIME1, [Materials]MatName is at the file level and the [Materials]MatTime'MatTimeID is at the sub-file level. So sorting is not possible at the same level in STEEL-TIME1.

In STEEL-TIME1, there are three files and [Activities] is a separate file related to the [Facilities] file. From a developer's point of view, STEEL-TIME1 is more complicated than STEEL-TIME2. Moreover, most of the searches are performed at the sub-file level. For example, it is possible to list the dual prices and the reduced cost coefficients in the output layout at the file level in STEEL-TIME2, but similar lists are not possible in the

STEEL-TIME1. Such a display can be available in STEEL-TIME1 at the sub-record level only. On the other hand, STEEL-TIME1 has a greater flexibility for listing the activities, as [Activities] is a separate output file. Because of the inherent advantages of the relational file, the user will be able to update activities separately. Although we have not implemented this concept in STEEL-TIME1, such an implementation is possible. STEEL-TIME1 will also allow the user to compare two activities of two facilities by listing activities on the output file. So an activity PRODUCTION OF ES1 in three facilities M1, M2, M3 can be listed by performing a search with [Activities]ActName = "PRODUCTION OF ES1". Such searches are not possible with STEEL-TIME2.

### **6.3 Data Storage and Retrieval**

STEEL-TIME1 satisfies the conditions of normalization that no piece of information be stored in more than one place. This condition is not satisfied in STEEL-TIME2. We also see that STEEL-TIME2 takes greater storage space than STEEL-TIME1.

In STEEL-TIME2 certain fields are repeated. [Materials]MatName, [Materials]MatType, [Materials]MatInvZero, [Materials]MatUnits, [Facilities]FacName and [Facilities]FacUnits are the fields that are repeated for every record of the [Time]TimeID file. This certainly requires more space for data storage, but does not pose a very serious problem with respect to ease of use. The 4th Dimension software allows a script to be written so that when the user enters the data for [Materials]MatName for one time period, the same [Materials]MatName is also available in other time periods. Therefore, as long as we are not changing [Materials]MatTime, we do not need to enter the data for each time period.

### **6.4 Ease of Development**

STEEL-TIME2 is easier to develop than STEEL-TIME1. However, we have decided to opt for STEEL-TIME1 as our main implementation, primarily because the latest version of the 4th Dimension software does not support more than one level of sub-file. Because of the inherent advantage of relational databases, [Activities] was defined as a separate file in STEEL-TIME1, whereas it was a sub-file in the [Facilities] file of STEEL-TIME2.

## 6.5 Efficiency

The times for constraint generation, variable generation and solution, and reading optimal values and the dual values are as shown in Table 5.

### Insert Table 5

We find that STEEL-TIME2 is faster in generating the variables and constraints than STEEL-TIME1. This is because in STEEL-TIME1, the data for time dependent parameters are stored in a sub-file. So every time a record is written in the [Variables] file, first the record of the [Materials] is searched for, then the sub-record of the file is searched for, and then the record is written in the [Variables] file. However in STEEL-TIME2 fields like BuyMax, BuyMin are at the field level. Therefore to write a record in the [Variables] file, we only have to search the [Materials] and [Facilities] at the file level. Similarly, the disk-space for the data of STEEL-TIME2 is higher than that of STEEL-TIME1. This is because time-independent parameters like MatName, MatTag, MatInvZero, FacName, FacTag, FacType etc. are duplicated in STEEL-TIME2.

After a careful comparison of these two variations, we find that STEEL-TIME2 is superior to STEEL-TIME1 on an overall basis. However we need to extend the present study so that STEEL-TIME2 is normalized. This can be done by replacing all the sub-files by files so that [Materials]MatTime and [Facilities]FacTime and other sub-files will be normalized with additional indices and key-fields. We will be in a position to recommend STEEL-TIME2 only after that.

## 7. Extension and Conclusion

An extension of the DSS will be non-linearity of the model. Most of the industrial cost curves are non-linear or at best can be represented as having a piece-wise linear behavior. It will be interesting to study how to represent these non-linearities while retaining the model's user-friendliness.

A second extension of the model will be to have multiple objective linear programs and represent them in the database. This can be done by changing the model management system. For example, the current model can be changed to cost minimization, revenue maximization, maximization of marketable products (revenue or

production), maximization of the utilization of the facilities etc. It is possible to have a menu driven program in this DSS which optimizes over different objectives.

An interesting extension will be to study the paradigm neutrality Geoffrion [10] of this data structure for the multi-period model. Although the model is designed for the mathematical programming paradigm, we can extend it for inventory control and also for scheduling, vehicle routing and queuing applications. We have parameters for all materials at all times. We can determine the ordering and holding cost for all material and hence try to find optimal order quantities. However, the batch size will be decided by practical considerations like the **heat size** (is this correct?) of the steel making shop, the capacity of the vehicle carrying the products and the capacity of the loading and unloading facility. Given that we have the batch size and lead-time of all materials produced, the present model can be extended to a scheduling model of each product in each time.

## Tables

**Table 1**

Correspondence of [Materials]File and the LP Model

Sr. No.	Parameter of the LP	Fields of the Tables of the Database
1	$l_{jt}^{buy}$	[Materials]MatTime'BuyMin
2	$u_{jt}^{buy}$	[Materials]MatTime'BuyMax
3	$c_{jt}^{buy}$	[Materials]MatTime'BuyPrice
4	$l_{jt}^{sell}$	[Materials]MatTime'SellMin
5	$u_{jt}^{sell}$	[Materials]MatTime'SellMax
6	$c_{jt}^{sell}$	[Materials]MatTime'SellPrice



7	$l_{jt}^{inv}$	[Materials]MatTime'InvMin
8	$u_{jt}^{inv}$	[Materials]MatTime'InvMax
9	$h_{jt}$	[Materials]MatTime'MatInvCCOST
10	$x_{j0}^{inv}$	[Materials]MatInvZero
11	$a_{jt}^{conv}$	[Materials]Conversion'ConvYield
12	$c_{jt}^{conv}$	[Materials]Conversion'ConvCost
13	$Comp_{j\alpha}^{\max}$	[Materials]Compositions'CompMax
14	$Comp_{j\alpha}^{\min}$	[Materials]Compositions' CompMin

**Table 2**

Correspondence of [Facilities] File and the LP Model

Sr. No.	Parameter of the LP	Fields of the Tables of the Database
1	$l_{ijt}^{in}$	[Facilities]Inputs'InMin
2	$u_{ijt}^{in}$	[Facilities]Inputs'InMax
3	$l_{ijt}^{out}$	[Facilities]Outputs'OutMin
4	$u_{ijt}^{out}$	[Facilities]Outputs'OutMax
5	$C_{it}^{vend}$	[Facilities]FacTime'Vendor_Cost
6	$l_{it}^{cap}$	[Facilities]FacTime'CapMin
7	$u_{it}^{cap}$	[Facilities]FacTime'CapMax

**Table 3**

Correspondence of [Activities] File and the LP Model

Sr. No.	Parameter of the LP	Fields of the Tables of the Database
1	$l_{ikt}^{act}$	[Activities]ActMin
2	$u_{ikt}^{act}$	[Activities]ActMax
3	$u_{ikt}^{act}$	[Activities]ActCost
4	$r_{ikt}^{act}$	[Activities]ActCapUsed
5	$a_{ijkt}^{in}$	[Activities]ActInputs'ActInMat
6	$a_{ijkt}^{out}$	[Activities]ActOutPuts'ActOutMat

**Table 4**

Correspondence of [Storages\_Areas] File and the LP Model

Sr. No.	Parameter of the LP	Fields of the Tables of the Database
1	$l_{st}$	[Storage-Areas]StoreMin
2	$u_{st}$	[Storage-Areas]StoreMax

**Table 5**

Comparison of Steel-Time 1 and Steel-Time2

Computer	Macintosh	
Database	STEEL-TIME1	STEEL-TIME2T-2
Records in Files		

<b>Materials</b>	<b>19</b>	<b>57</b>
<b>Facilities</b>	<b>21</b>	<b>21</b>
<b>Activities</b>	<b>24</b>	<b>24 (Sub-file)</b>
<b>Times</b>	<b>3</b>	<b>3</b>
<b>Constraints</b>	<b>141</b>	<b>141</b>
<b>Vairables</b>	<b>266</b>	<b>266</b>
<b>Disk Space (Model)</b>	<b>688</b>	<b>336</b>
<b>Disk Space (Data)</b>	<b>472</b>	<b>484</b>
<b>Cons. Generation Time</b>	<b>12</b>	<b>12</b>
<b>Var.Generation Time</b>	<b>109</b>	<b>45</b>
<b>Writing Constraint Time</b>	<b>7</b>	<b>7</b>
<b>Writing Variable Time</b>	<b>22</b>	<b>21</b>
<b>Solving</b>	<b>8</b>	<b>8</b>
<b>Reading Optimal Value Tim</b>	<b>21</b>	<b>21</b>
<b>Reading Dual Value Time</b>	<b>8</b>	<b>8</b>

### **Acknowledgement**

This work has been supported in part by grants from the American Iron and Steel Institute and its members, by grant DDM-8908818 from the U.S. National Science Foundation, and by the Research and Publication Committee of the Indian Institute of Management, Ahmedabad.

## Appendix

### Model Formulation

We first define the data, in five parts: times, materials, facilities, activities, and storage-areas. The notation for the decision variables is then presented. Finally the objective and constraints are described, in both words and formulae.

All quantities of materials are taken to be in the same units, such as kilograms.

#### **Time data**

$T = \{1, \dots, T\}$  is the set of time periods in the planning horizon, indexed by  $t$

$\rho$  is the interest rate per period, taken as zero if there is no discounting

#### **Materials data**

$M$  is the set of all materials

$l_{jt}^{buy}$  = lower limit on purchases of material  $j$ , for each  $j \in M$  and  $t \in T$

$u_{jt}^{buy}$  = upper limit on purchases of material  $j$ , for each  $j \in M$  and  $t \in T$

$C_{jt}^{buy}$  = cost per unit of material  $j$  purchased, for each  $j \in M$  and  $t \in T$

$l_{jt}^{sell}$  = lower limit on sales of material  $j$ , for each  $j \in M$  and  $t \in T$

$u_{jt}^{sell}$  = upper limit on sales of material  $j$ , for each  $j \in M$  and  $t \in T$

$C_{jt}^{sell}$  = revenue per unit of material  $j$ , for each  $j \in M$  and  $t \in T$

$l_{jt}^{inv}$  = lower limit on inventory of material  $j$ , for each  $j \in M$  and  $t \in T$

$u_{jt}^{inv}$  = upper limit on inventory of material  $j$ , for each  $j \in M$  and  $t \in T$

$V_{j0}^{inv}$  = initial inventory of material  $j$ , for each  $j \in M$

$C_{jt}^{inv}$  = holding cost per unit of material  $j$ , for each  $j \in M$  and  $t \in T$

$M^{conv} \subseteq \{j \in M, j' \in M : j \neq j'\}$  is the set of conversions:

$(j, j') \in M^{conv}$  means that material  $j$  can be converted to material  $j'$

$\alpha_{jj't}^{conv}$  = number of units of material  $j'$  that result from converting one unit of material  $j$ , for each

$(j, j') \in M^{conv}, t \in T$

$C_{jj't}^{conv}$  = cost per unit of material  $j$  of the conversion from  $j$  to  $j'$ , for each  $(j, j') \in M^{conv}, t \in T$

## Facilities data

$F$  is the set of facilities

$l_{it}^{cap}$  = the minimum amount of the capacity of facility  $i$  that must be used, for each  $i \in F$  and  $t \in T$

$u_{it}^{cap}$  = the capacity of facility  $i$ , for each  $i \in F$  and  $t \in T$

$C_{it}^{cap}$  = the cost of vendoring (outsourcing) a unit of capacity at facility  $i$ , for each  $i \in F$  and  $t \in T$

$F^{in} \subseteq F \times M$  is the set of facility inputs:

$(i, j) \in F^{in}$  means that material  $j$  is used as an input at facility  $i$

$l_{ijt}^{in}$  = the minimum amount of material  $j$  that must be used as input to facility  $i$ , for each  $(i, j) \in F^{in}, t \in T$

$u_{ijt}^{in}$  = the maximum amount of material  $j$  that must be used as input to facility  $i$ , for each  $(i, j) \in F^{in}, t \in T$

$F^{out} \subseteq F \times M$  is the set of facility outputs:

$(i, j) \in F^{out}$  means that material  $j$  is produced as an output at facility  $i$

$l_{ijt}^{out}$  = the minimum amount of material  $j$  that must be produced as output at facility  $i$ , for each  
 $(i,j) \in F^{out}, t \in T$

$u_{ijt}^{out}$  = the maximum amount of material  $j$  that must be produced as output at facility  $i$ , for each  
 $(i,j) \in F^{out}, t \in T$

### Activities data

$F^{act} \subseteq \{(i,k) : i \in F\}$  is the set of activities:

$(i,k) \in F^{act}$  means that  $k$  is an activity available at facility  $i$

$l_{ikt}^{act}$  = the minimum number of units of activity  $k$  that may be run at facility  $i$ , for each  $(i,k) \in F^{act}, t \in T$

$u_{ikt}^{act}$  = the maximum number of units of activity  $k$  that may be run at facility  $i$ , for each  $(i,k) \in F^{act}, t \in T$

$c_{ikt}^{act}$  = the cost per unit of running activity  $k$  at facility  $i$ , for each  $(i,k) \in F^{act}, t \in T$

$r_{ikt}^{act}$  = the number of units of activity that can be accommodated in one unit of capacity of facility  $i$ ,  
for each  $(i,k) \in F^{act}, t \in T$

$A^{in} \subseteq \{(i,j,k,t) : (i,j) \in F^{in}, (i,k) \in F^{act}, t \in T\}$  is the set of activity inputs:

$(i,j,k,t) \in A^{in}$  means that input material  $j$  is used by activity  $k$  at facility  $i$  during time period  $t$

$\alpha_{ijk}^{in}$  = units of input material  $j$  required by one unit of activity  $k$  at facility  $i$  in time period  $t$ , for each

$(i,j,k,t) \in A^{in}$

$A^{out} \subseteq \{(i,j,k,t) : (i,j) \in F^{out}, (i,k) \in F^{act}, t \in T\}$  is the set of activity outputs:

$(i,j,k,t) \in A^{out}$  means that output material  $j$  is produced by activity  $k$  at facility  $i$  during time period  $t$

$\alpha_{ijk}^{out}$  = units of output material  $j$  produced by one unit of activity  $k$  at facility  $i$  in time period  $t$ , for each  $(i,j,k,t) \in A^{out}$

### Storage-areas data

$S$  is the set of storage areas

$l_{st}^{stor}$  = lower limit on total material in storage area  $s$ , for each  $s \in S, t \in T$

$u_{st}^{stor}$  = upper limit on total material in storage area  $s$ , for each  $s \in S, t \in T$

### Variables

$x_{jt}^{buy}$  = units of material  $j$  bought, for each  $j \in M, t \in T$

$x_{jt}^{sell}$  = units of material  $j$  sold, for each  $j \in M, t \in T$

$x_{jst}^{stor}$  = units of material  $j$  in storage area  $s$ , for each  $j \in M, s \in S, t \in T$

$x_{jt}^{inv}$  = total units of material  $j$  in inventory (storage), for each  $j \in M, t \in T$

$x_{j0}^{inv}$  = initial inventory of material  $j$ , for each  $j \in M$

$x_{jj't}^{conv}$  = units of material  $j$  converted to material  $j'$ , for each  $(j,j') \in M^{conv}, t \in T$

$x_{ijt}^{in}$  = units of material  $j$  used as input by facility  $i$ , for each  $(i,j) \in F^{in}, t \in T$

$x_{ijt}^{out}$  = units of material  $j$  produced as output by facility  $i$ , for each  $(i,j) \in F^{out}, t \in T$

$x_{ikt}^{act}$  = units of activity  $k$  operated at facility  $i$ , for each  $(i,k) \in F^{act}, t \in T$

$x_{it}^{cap}$  = units of capacity vendored at facility  $i$ , for each  $i \in F$ ,  $t \in T$

## Objective

Maximize the sum, over all time periods, of revenues from sales less costs of purchasing, holding inventories, converting, operating activities at facilities and vendoring:

$$\sum_{t \in T} (1 + \rho)^{-t} Z(t)$$

Where,

$$\begin{aligned} Z(t) = & \sum_{j \in M} c_{jt}^{sell} x_{jt}^{sell} - \sum_{j \in M} c_{jt}^{buy} x_{jt}^{buy} - \sum_{j \in M} c_{jt}^{inv} x_{jt}^{inv} - \sum_{(j,j') \in M^{conv}} c_{jj't}^{conv} x_{jj't}^{conv} - \sum_{(i,k) \in F^{act}} c_{ikt}^{act} x_{ikt}^{act} \\ & - \sum_{i \in F} c_{it}^{cap} x_{it}^{cap} \end{aligned}$$

## Constraints

For each  $j \in M$ ,  $r \in R$  and  $t \in T$ , the amount of material  $j$  made available by purchases, production, conversions and beginning inventory must equal the amount used for sales, production, conversions and ending inventory:

$$\begin{aligned} x_{jt}^{sell} + \sum_{(i,j) \in F^{out}} x_{ijt}^{out} + \sum_{(j',j) \in M^{conv}} \alpha_{j'jt}^{conv} x_{j'jt}^{conv} + x_{jt-1}^{inv} = & x_{jt}^{sell} + \sum_{(i,j) \in F^{in}} x_{ijt}^{in} + \sum_{(j,j') \in M^{conv}} x_{jj't}^{conv} \\ & + x_{jt}^{inv} \end{aligned}$$

For each  $(i,j) \in F^{in}$  and  $t \in T$ , the amount of input  $j$  used at facility  $i$  must equal the total consumption by all the activities at facility  $i$ :

$$x_{ijt}^{in} = \sum_{(i,j,k,t) \in A^{in}} \alpha_{ijkt}^{in} x_{ikt}^{act}$$



For each  $(i,j) \in F^{out}$  and  $t \in T$ , the amount of output  $j$  produced at facility  $i$  must equal the total production by all the activities at facility  $i$ :

$$x_{ijt}^{out} = \sum_{(i,j,k,t) \in A^{out}} \alpha_{ijkt}^{out} x_{ikt}^{act}$$

For each  $i \in F$  and  $t \in T$ , the capacity used by all activities at facility  $i$  must be within the range given by the lower limit and the upper limit plus the amount of capacity vendored:

$$l_{it}^{cap} \leq \sum_{(i,k) \in F^{act}} x_{ikt}^{act} / r_{ikt}^{act} \leq u_{it}^{cap} + x_{it}^{cap}$$

For each  $j \in M$ , the amount of material inventoried in the plant before the first time period is defined to equal the specified initial inventory:

$$x_{j0}^{inv} = v_{j0}$$

For each  $j \in M$  and  $t \in T$ , the total amount of material  $j$  inventoried is defined as the sum of the inventories over all storage areas:

$$\sum_{s \in S} x_{jst}^{stor} = x_{jt}^{inv}$$

For each  $s \in S$  and  $t \in T$ , the total of all materials inventoried in storage area  $s$  must be within the specified limits:

$$l_{st}^{stor} \leq \sum_{j \in M} x_{jst}^{stor} \leq u_{st}^{stor}$$

All variables must lie within the relevant limits defined by the data:

$$l_{jt}^{buy} \leq x_{jt}^{buy} \leq u_{jt}^{buy}, \quad \text{for each } j \in M \text{ and } t \in T$$

$$l_{jt}^{sell} \leq x_{jt}^{sell} \leq u_{jt}^{sell}, \quad \text{for each } j \in M \text{ and } t \in T$$

$$l_{jt}^{inv} \leq x_{jt}^{inv} \leq u_{jt}^{inv}, \quad \text{for each } j \in M \text{ and } t \in T$$

$$0 \leq x_{jj't}^{conv}, \quad \text{for each } (j,j') \in M^{conv} \text{ and } t \in T$$

$$0 \leq x_{it}^{cap}, \quad \text{for each } i \in F \text{ and } t \in T$$

$$0 \leq x_{jst}^{stor}, \quad \text{for each } s \in S, j \in M \text{ and } t \in T$$

$$l_{ij}^{in} \leq x_{ij}^{in} \leq u_{ij}^{in}, \quad \text{for each } (i,j) \in F^{in} \text{ and } t \in T$$

$$l_{ij}^{out} \leq x_{ij}^{out} \leq u_{ij}^{out}, \quad \text{for each } (i,j) \in F^{out} \text{ and } t \in T$$

$$l_{ik}^{act} \leq x_{ik}^{act} \leq u_{ik}^{act}, \quad \text{for each } (i,j) \in F^{act} \text{ and } t \in T$$

## References

- [1] Adams, D., Beckett, D., 1999. Programming in 4<sup>th</sup> Dimension, the Ultimate Guide. Automated Solutions Group: Huntington Beach, CA, USA.
- [2] Date, C. J. (1981), Introduction to database systems, 3<sup>rd</sup> Edition, Addison-Wesley Publishing Company
- [3] Dolk, D. R. (1986), Data as Models: An Approach to Implementing Model Management, *Decision Support Systems*, 2 (1)
- [4] Dolk, D. R. (1986a), A Generalized Model Management System for Mathematical Programming, *ACM Transactions on Mathematical Software*, 12 (2), pp 92 -126
- [5] Dominguez – Ballesteros, B., Mitra, G., Lucas, C., & Koutsoukis, N- S. (2002), Modeling and Solving Environments for Mathematical Programming (MP): A Status Review and New Directions, *Journal of Operational Research Society*, 53, pp1072 – 1092
- [6] Dutta G., & Fourer R., A survey of mathematical programming applications in an integrated steel plant, *Manufacturing & Service Operations Management*, 3(4) (2001), 387-400.

- [7] Dutta G., & Sinha G. P., & Roy P.N, and Mitter N., A linear programming model for distribution of electrical energy in a steel plant, *International Transactions in Operational Research*, 1(1) (1994) 17-29.
- [8] Fourer, R. (1983), Modeling Language Versus Matrix generator for Linear Programming, *ACM Transactions on Mathematical Software*, 9 (2), pp 143 – 183
- [9] Fourer R., Database structure for mathematical programming models, *Decision Support Systems*, 20 (1997) 317-344.
- [10] Geoffrion A. M., Computer-based modeling environments, *European Journal in Operations Research*, 41 (1989) 33-43.
- [11] Marsten R.E., The design of XMP linear programming library, *ACM Transactions on Mathematical Software*, 7 (1981) 481- 497.
- [12] Murphy, F. H., Storh, E. A., & Asthana, A (1992), Representation Scheme for Linear Programming Models, *Management Science*, 38 (7), pp 964 – 991
- [13] SCH Leung, Y. Wu., & Lai K. K. (2006), A Stochastic Programming Approach for Multi-Site Aggregate Production Planning, *Journal of Operational Research Society*, 57, 123-132
- [14] Sinha G. P., Chandrashekaran B.S., Mitter N., Dutta G., Singh S.B., Roy P.N., & Roy Choudhary A., Strategic and operations management with optimization at Tata Steel, *Interfaces*, 25 (1) (1995) 6-19.

Figures

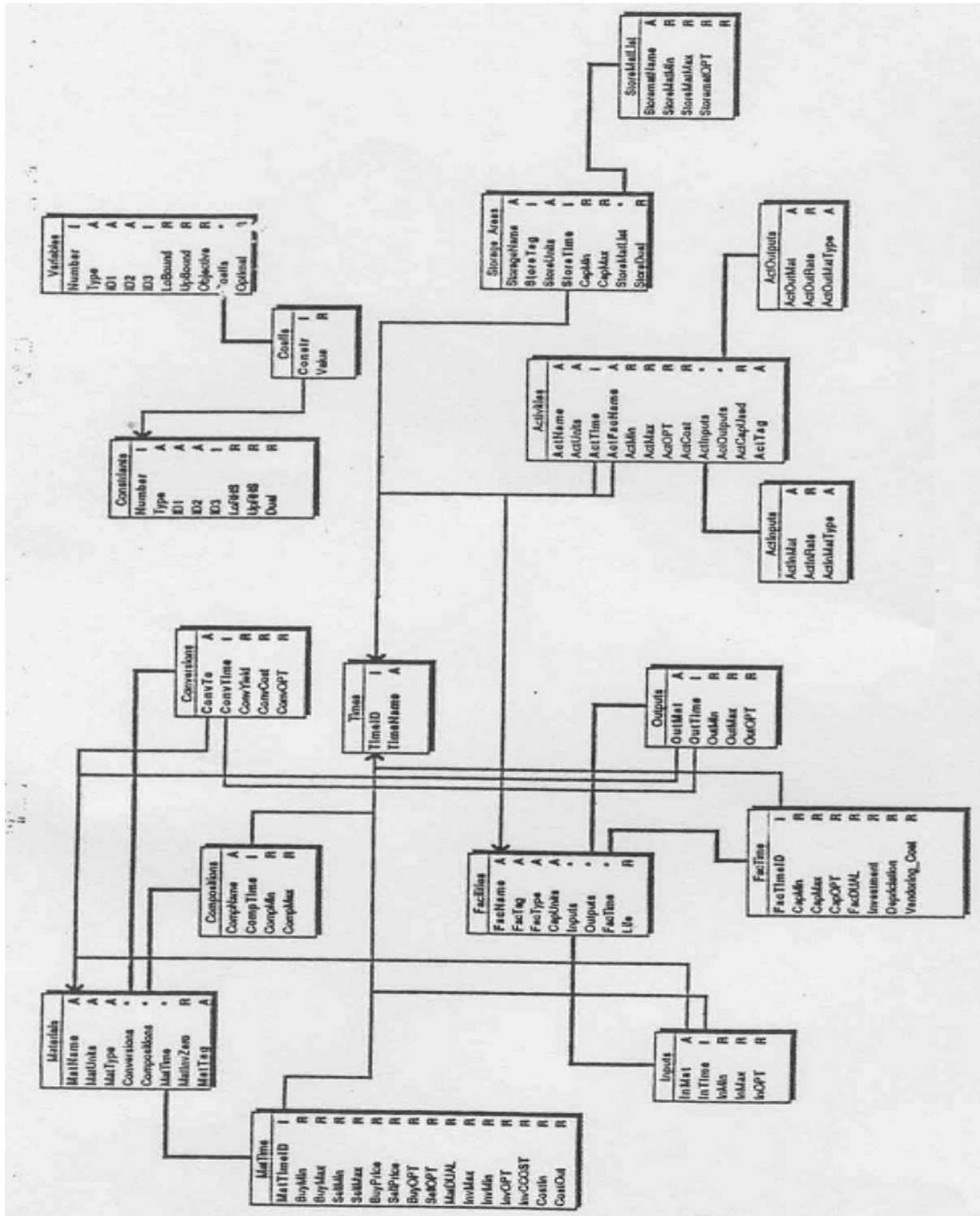


Figure 1: Database structure for STEEL-TIME

Time Input	
<b>Save</b>  1/3  <b>Next</b>  <b>Previous</b>  <b>Cancel</b>  <b>Delete</b>	<b>TimeID</b>   1 _____  <b>TimeName</b>   JAN _____

Figure 2: Time Input

Custom			
MAT-TAG	MATERIAL	UNITS	MATERIAL TYPE
101	CC BILLET	TONS	Inter mediate
102	SINTER	TONS	Input
105	SLAG	TONS	Inter mediate
106	HOT METAL	TONS	Inter mediate
110	ORE	TONS	Input
201	COAL	TONS	Input
202	COKE	TONS	Inter mediate
213	NEW MAT	TONS	Input
401	CRUDE STEEL	TONS	Input
402	STEEL for CC	TONS	Inter mediate
501	HEAVY MELTING SCRAP	TONS	Inter mediate
503	STEEL SCRAP	TONS	Inter mediate
503	LIGHT MELTING SCRAP	TONS	Inter mediate
504	MILL SCRAP	TONS	Inter mediate
601	BILLET	TONS	Inter mediate
602	BLOOM	TONS	Inter mediate
607	SLAB	TONS	Inter mediate

**Add Entry**
**Done**

Figure 3: Output Layout of Materials File

**Custom**

---

**Save**

3/19

**Next**

**Previous**

**Cancel**

**Delete**

**SORT TIME**

**Materials Input**      MatTag

---

**Name**       **Units**

**Type**       **Initial Inventory**

*Conversions*

Conv. Material	Time	Yield	Cost
HEAVY MELTING SCRAP	1	0.812	90
HEAVY MELTING SCRAP	2	0.8	82

*Time*

Time	BuyMax	SellMax	InvMax	BuyPrice	SellPrice
1	0	200000	200000	121	124
2	0	30900	2000	123	124

*Compositions*

Time	Name	Minimum	Maximum
1	C	1.2	1.3
1	Si	0.69	0.75

Figure 4: Input Layout of Materials File

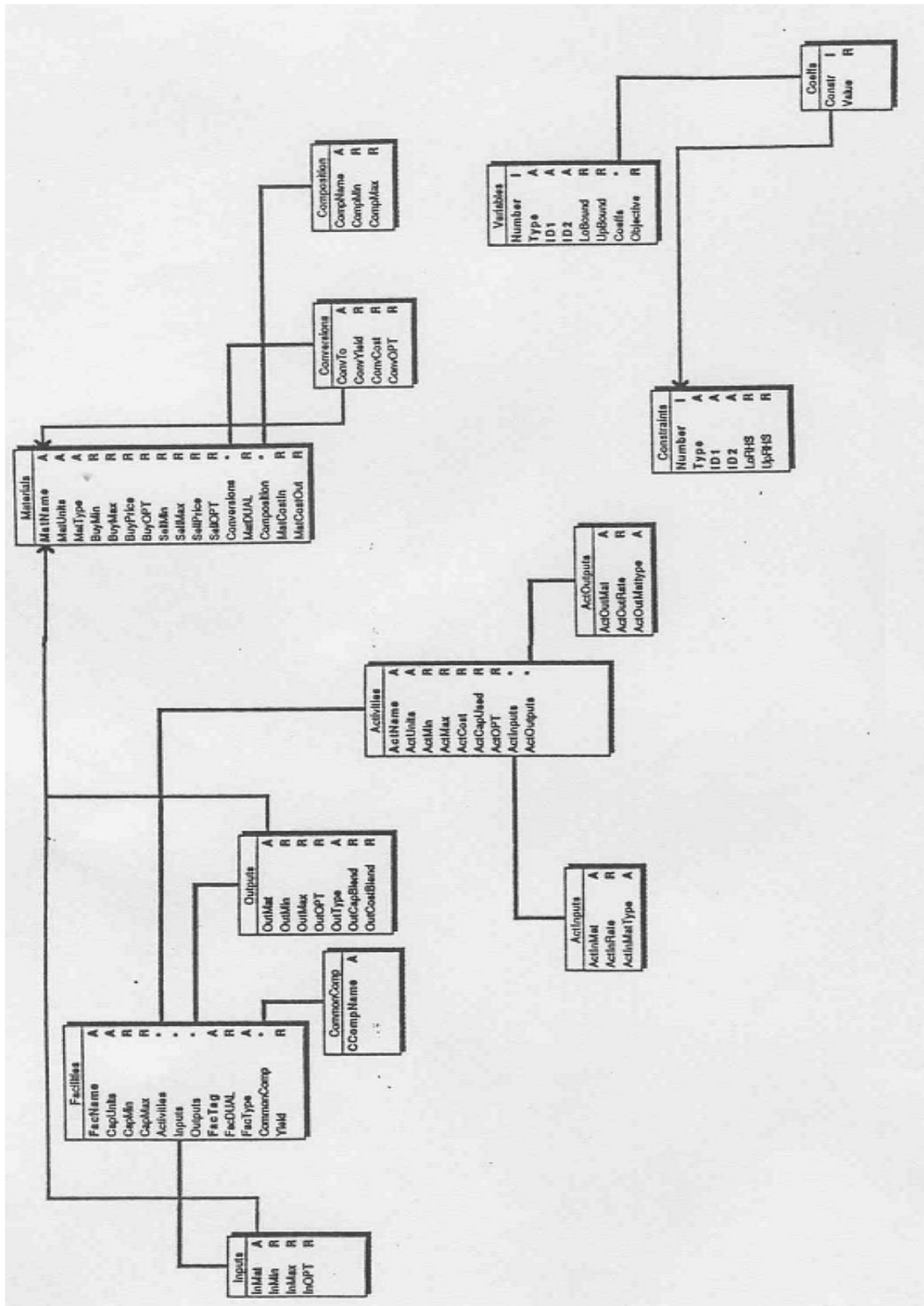


Figure 5: Database structure for STEEL

Facilities Input					
<b>Save</b> 3/7 <b>Next</b> <b>Previous</b> <b>Cancel</b> <b>Delete</b>  <b>Activities</b> <b>SORT TIME</b> <b>SORT MATL</b>	<b>Name</b>	BASIC OXYGEN FURNACE	<b>FacTag</b>	0003	
	<b>FacType</b>	PRODUCT-MIX	<b>Units</b>	TONS	
	<i>Inputs</i>				
	<b>Material Input</b>		<b>Time</b>	<b>Minimum</b>	<b>Maximum</b>
	HOT METAL		3	0	9999999
	HOT METAL		2	0	999999
	<i>Outputs</i>				
	<b>Material Output</b>		<b>Time</b>	<b>Minimum</b>	<b>Maximum</b>
	CRUDE STEEL		3	0	9999999
	CRUDE STEEL		2	0	9999999
<i>Time</i>					
<b>Time</b>	<b>CapMin</b>	<b>CapMax</b>			
1	0	670			
2	0	644			

Figure 6: Input Layout of Facilities File

FACILITY NAME	FAC TAG	FAC TYPE	CAPACITY UNITS
BLAST FURNACE	0001	PRODUCT-MIX	TONS
COKE OVENS	0002	PRODUCT-MIX	TONS
BASIC OXYGEN FURNACE	0003	PRODUCT-MIX	TONS
CONTINUOUS CASER	0004	PRODUCT-MIX	TONS
ROLLING MILL NO 1	0005	PRODUCT-MIX	HOURS
MERCHANT MILL NO. 1	0006	PRODUCT-MIX	TONS
S.B.B. MILL 1	0007	PRODUCT-MIX	TONS

Add Entry
Done

Figure 7: Output Layout of Facilities File



Activities		Tag	009
<b>Save</b>	<b>Name</b> CC STEEL PRODN	<b>Units</b>	TONS
<b>Next</b>	<b>Time</b> 2	<b>Cost</b>	120
<b>Previous</b>	<b>Facility</b> BASIC OXYGEN FURNACE		
<b>Cancel</b>	<b>Minimum</b> 0	<b>Maximum</b>	100008
<b>Delete</b>	<b>Use/Unit Facility Capacity</b>		651
<i>Inputs</i>			
ActInMat	ActInRate		
HOT METAL	1		
STEEL SCRAP	0.05		
<i>Outputs</i>			
ActOutMat	ActOutRate		
CRUDE STEEL	1		

Figure 8: Input Layout of Activities File

Activity Name	Time	Minimum	Maximum	Cost	ActTag
BILLET PRODN AT SBEM	3	0	67000	120	024
BILLET PRODN AT SBEM	2	0	95000	67	023
BILLET PRODN AT SBEM	1	0	40000	45	022
WIRE RODS PRODN	3	0	30000	45	021
WIRE RODS PRODN	2	0	30000	34	020
WIRE RODS PRODN	1	0	30000	23	019
PRDDN OF BLOOM	3	0	98000	23	018
PRDDN OF BLOOM	2	0	92000	19	017
PRDDN OF BLOOM	1	0	100000	43	016
CC BILLET PRODN	3	0	50000	123	015
CC BILLET PRODN	2	0	60000	89	014
CC BILLET PRODN	1	0	50000	67	013
CRUDE STEEL PRODN	3	0	9999999	123	012
CC STEEL PRODN	3	0	100000	11	011
CRUDE STEEL PRODN	2	0	9999999	123	010
CC STEEL PRODN	2	0	100008	120	009
CRUDE STEEL PRODN	1	0	9999999	12	008
CC STEEL PRODN	1	0	120000	10	007
PRDDN OF COKE	3	0	9999999	120	006
PRDDN OF COKE	2	0	9999999	56	005
PRDDN OF COKE	1	0	9999999	45	004

Figure 9: Output Layout of Activities File

Storage-Areas Input										
<b>Save</b>	<b>Name</b> RAW MATERIAL STORE									
2/3	<b>Tag</b> 10002									
<b>Next</b>	<b>Units</b> TONS									
<b>Previous</b>	<b>Time</b> 1									
<b>Cancel</b>	<b>CapMin</b> 0									
<b>Delete</b>	<b>CapMax</b> 45000									
<table border="1"> <thead> <tr> <th>Material Name</th> <th>Minimum</th> <th>Maximum</th> </tr> </thead> <tbody> <tr> <td>COAL</td> <td>0</td> <td>4560</td> </tr> <tr> <td>BILLET</td> <td>0</td> <td>27000</td> </tr> </tbody> </table>		Material Name	Minimum	Maximum	COAL	0	4560	BILLET	0	27000
Material Name	Minimum	Maximum								
COAL	0	4560								
BILLET	0	27000								

Figure 10: Input Layout of Storage-Areas File

Custom					
StorageName	StoreTag	StoreTime	StoreUnits	CapMin	CapMax
FINISHED GOOD STORE	10001	1	TONS	0	27000
RAW MATERIAL STORE	10002	1	TONS	0	45000
STORE AT OPEN SPACE	10004	3	TONS	0	80000

**Add Entry** **Done**

Figure 11: Output Layout of Storage-Areas File

		Variables															
<input type="button" value="Done"/>  <input type="button" value="Next"/>  <input type="button" value="Previous"/>	<b>Number</b>	<input type="text" value="104"/>															
	<b>Type</b>	<input type="text" value="Material Sold"/>	<b>LoBound</b>	<input type="text" value="0"/>													
	<b>ID1</b>	<input type="text" value="CC BILLET"/>	<b>UpBound</b>	<input type="text" value="99999"/>													
	<b>ID2</b>	<input type="text"/>															
	<b>ID3/Time</b>	<input type="text" value="1"/>	<b>Optimal</b>	<input type="text" value="0"/>													
	<b>Objective</b>	<input type="text" value="5,000.00"/>															
	<table border="1"> <thead> <tr> <th>Rev</th> <th>Value</th> <th>Time</th> <th>Type</th> <th>ID1</th> <th>ID2</th> </tr> </thead> <tbody> <tr> <td>24</td> <td></td> <td>1</td> <td>Material Balance</td> <td>CC BILLET</td> <td></td> </tr> </tbody> </table>	Rev	Value	Time	Type	ID1	ID2	24		1	Material Balance	CC BILLET					
Rev	Value	Time	Type	ID1	ID2												
24		1	Material Balance	CC BILLET													

Figure 12: Input Layout of Variables File

		Constraints			
<input type="button" value="Done"/>  <input type="button" value="Next"/>  <input type="button" value="Previous"/>	<b>Number</b>	<input type="text" value="127"/>			
	<b>Type</b>	<input type="text" value="Facility Capacity"/>	<b>LoRHS</b>	<input type="text" value="0"/>	
	<b>ID1</b>	<input type="text" value="CONTINUOUS CASTER"/>	<b>UpRHS</b>	<input type="text" value="50000"/>	
	<b>ID2</b>	<input type="text"/>			
	<b>ID3/Time</b>	<input type="text" value="1"/>	<b>Dual</b>	<input type="text" value="0"/>	

Figure 13: Layout of Constraints File

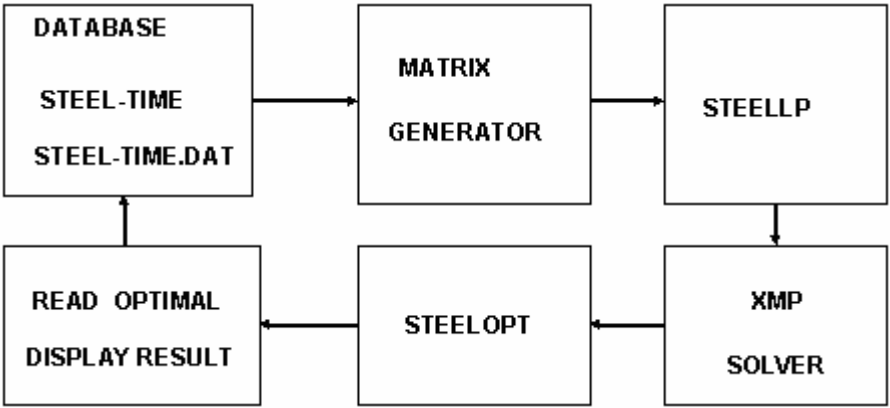


Figure 14: Optimization Steps

Time Material				
Save 1/3 Next Previous Cancel Delete	TimeID	<input type="text" value="1"/>		
	TimeName	<input type="text" value="JAN 1997"/>		
	<b>Material Name</b>	<b>Type</b>	<b>Units</b>	<input type="button" value="↑"/>
	BILLET	TONS	Intermediate	<input type="button" value="↓"/>
	CRUDE STEEL	TONS	Input	
	SINTER	TONS	Input	
ORE	TONS	Input		
STEEL SCRAP	TONS	Intermediate		
SLABS	TONS	Intermediate		
WIRE RODS	TONS	Output		

Figure 15: Materials List in Time Layout

Materials Optimum																				
<input type="button" value="Save"/> 18/19 <input type="button" value="Next"/> <input type="button" value="Previous"/> <input type="button" value="Cancel"/> <input type="button" value="Delete"/>	<b>Name</b> <input type="text" value="WIRE RODS"/>																			
	<b>Units</b> <input type="text" value="TONS"/>																			
	<b>Type</b> <input type="text" value="Output"/>																			
	<b>Initial Inventory</b> <input type="text" value="1200"/>																			
	<i>Conversions</i>																			
	<table border="1"> <thead> <tr> <th>Time</th> <th>Converted To</th> <th>Cost</th> <th>ConvOPT</th> <th>ConvYield</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	Time	Converted To	Cost	ConvOPT	ConvYield														
Time	Converted To	Cost	ConvOPT	ConvYield																
<i>Time</i>																				
<table border="1"> <thead> <tr> <th>Time</th> <th>BuyOPT</th> <th>SellOPT</th> <th>DUAL</th> <th>InvOPT</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>12009</td> <td>9000</td> <td>0</td> </tr> <tr> <td>2</td> <td>100000</td> <td>0</td> <td>9090</td> <td>100000</td> </tr> <tr> <td>3</td> <td>0</td> <td>100000</td> <td>10000</td> <td>0</td> </tr> </tbody> </table>	Time	BuyOPT	SellOPT	DUAL	InvOPT	1	0	12009	9000	0	2	100000	0	9090	100000	3	0	100000	10000	0
Time	BuyOPT	SellOPT	DUAL	InvOPT																
1	0	12009	9000	0																
2	100000	0	9090	100000																
3	0	100000	10000	0																

Figure 16: Materials Optimal

Mat Time Optimal																									
<input type="button" value="Save"/> <input type="button" value="Next"/> <input type="button" value="Previous"/> <input type="button" value="Cancel"/> <input type="button" value="Delete"/>	<b>MatTime</b> <input type="text" value="1"/>																								
	<table border="0"> <tr> <td></td> <td style="text-align: center;"><b>Buy</b></td> <td style="text-align: center;"><b>Sell</b></td> <td style="text-align: center;"><b>Inv</b></td> </tr> <tr> <td><b>Min</b></td> <td><input type="text" value="0"/></td> <td><input type="text" value="0"/></td> <td><input type="text" value="0"/></td> </tr> <tr> <td><b>Max</b></td> <td><input type="text" value="100000"/></td> <td><input type="text" value="100000"/></td> <td><input type="text" value="100000"/></td> </tr> <tr> <td><b>Price</b></td> <td><input type="text" value="9000"/></td> <td><input type="text" value="9000"/></td> <td></td> </tr> <tr> <td><b>Optimal</b></td> <td><input type="text" value="0"/></td> <td><input type="text" value="12009"/></td> <td><input type="text" value="0"/></td> </tr> <tr> <td><b>DUAL</b></td> <td><input type="text" value="9000"/></td> <td></td> <td></td> </tr> </table>		<b>Buy</b>	<b>Sell</b>	<b>Inv</b>	<b>Min</b>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<b>Max</b>	<input type="text" value="100000"/>	<input type="text" value="100000"/>	<input type="text" value="100000"/>	<b>Price</b>	<input type="text" value="9000"/>	<input type="text" value="9000"/>		<b>Optimal</b>	<input type="text" value="0"/>	<input type="text" value="12009"/>	<input type="text" value="0"/>	<b>DUAL</b>	<input type="text" value="9000"/>		
		<b>Buy</b>	<b>Sell</b>	<b>Inv</b>																					
	<b>Min</b>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>																					
	<b>Max</b>	<input type="text" value="100000"/>	<input type="text" value="100000"/>	<input type="text" value="100000"/>																					
	<b>Price</b>	<input type="text" value="9000"/>	<input type="text" value="9000"/>																						
<b>Optimal</b>	<input type="text" value="0"/>	<input type="text" value="12009"/>	<input type="text" value="0"/>																						
<b>DUAL</b>	<input type="text" value="9000"/>																								
<b>Sensitivity</b>																									
<input type="button" value="Buy?"/>																									
<input type="button" value="Sell?"/>																									
<input type="button" value="Inv?"/>																									

Figure 17: MatTime Optimal

Material at JAN 1997				
<input type="button" value="Save"/> 7/19 <input type="button" value="Next"/> <input type="button" value="Previous"/> <input type="button" value="Cancel"/>	<b>Name</b>	SINTER		
	<b>Type</b>	Input		
	<b>Units</b>	TONS		
	<b>InvZero</b>	10000		
		<b>Buy</b>	<b>Sell</b>	<b>Inventory</b>
	<b>Minimum</b>	0.0	0.0	0.0
	<b>Optimal</b>	0.0	10000.0	0.0
	<b>Maximum</b>	99999.0	99999.0	9999
	<b>Price</b>	2200.0	2200.0	15.0
	<b>Sensitivity</b>	<input type="button" value="Buy?"/>	<input type="button" value="Sell?"/>	<input type="button" value="Inu?"/>
<b>Dual Price</b>	2200.0			

Figure 18: Material Input in Time Layout

Time Facilities				
<input type="button" value="Save"/> 2/3 <input type="button" value="Next"/> <input type="button" value="Previous"/> <input type="button" value="Cancel"/> <input type="button" value="Delete"/>	<b>TimeID</b>	2		
	<b>Name</b>	FEB 1997		
		<b>FacTag</b>	<b>FacName</b>	<b>FacType</b>
		0001	BLAST FURNACE	PRODUCT-MIX
		0002	COKE OVENS	PRODUCT-MIX
		0003	BASIC OXYGEN FURNACE	PRODUCT-MIX
		0005	ROLLING MILL NO 1	PRODUCT-MIX
		0006	MERCHANT MILL NO. 1	PRODUCT-MIX
	0004	CONTINUOUS CASTER	PRODUCT-MIX	
	0007	S.B.B. MILL 1	PRODUCT-MIX	

Figure 19: Facilities List in Time Layout

Facilities at JAN 1997		FacTag	0005	
<input type="button" value="Save"/> <input type="button" value="Next"/> <input type="button" value="Previous"/> <input type="button" value="Cancel"/> <input type="button" value="Graphs"/>	<b>Name</b>	ROLLING MILL NO 1	<i>Capacity</i>	
	<b>Type</b>	PRODUCT-MIX	<i>Minimum</i>	
	<b>Units</b>	HOURS	<i>Optimum</i>	
			<i>Maximum</i>	
			645.0	
	<i>Inputs</i>		<b>Dual?</b>	
			0.00	
	<b>Input Material</b>	<b>Minimum</b>	<b>Optimal</b>	<b>Maximum</b>
	CRUDE STEEL	0	0.00	100000.0
	<i>Outputs</i>			
<b>Output Material</b>	<b>Minimum</b>	<b>Optimal</b>	<b>Maximum</b>	
BLOOM	0	0	95000	
MILL SCRAP	0	0	30000	
<i>Activities</i>				
<b>ActName</b>	<b>Minimum</b>	<b>Optimal</b>	<b>Maximum</b>	<b>Cost</b>
PRDGN OF BLOOM	0	0	100000	43

Figure 20: Facilities in Time Layout

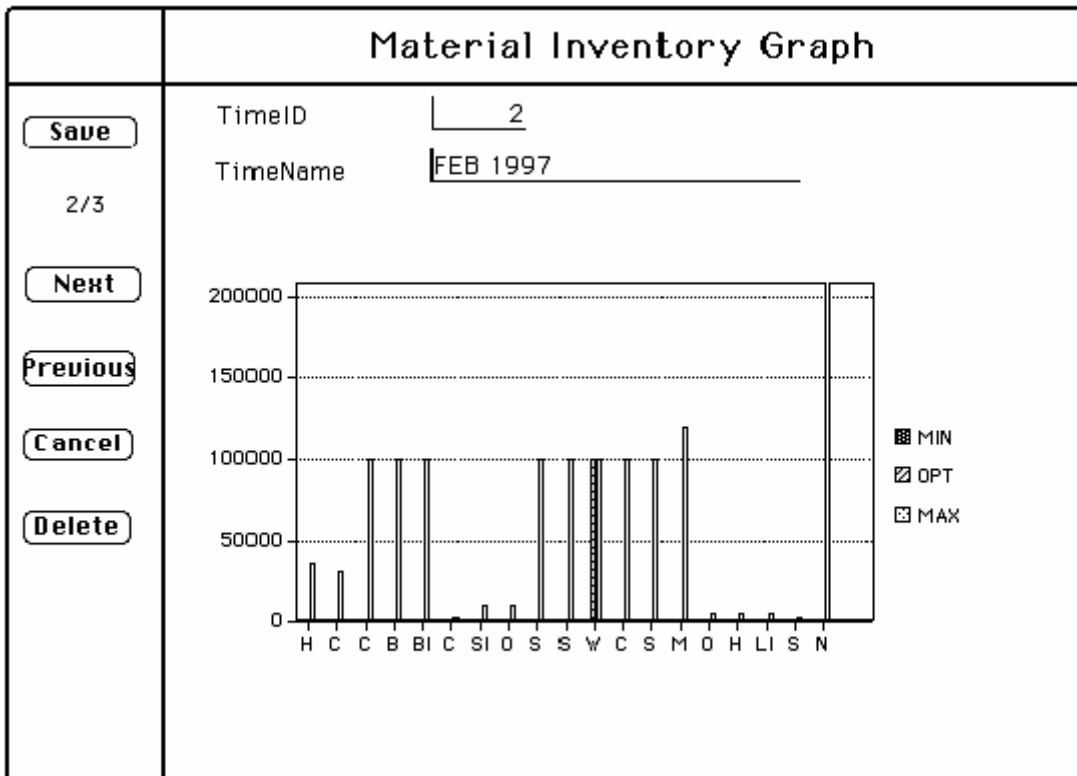


Figure 21: Graph of Material Inventoried

**Profit will improve by up to \$1,931.00  
for each unit decrease in the  
purchase minimum.**

**Profit will decline by at least  
\$1,931.00 for each unit increase in  
the purchase minimum.**

**OK**

Figure 22: Display of Dual Variables

<b>Grand Summary</b>	
<b>Revenue from Sales</b>	2,194,681,608.05
<b>Cost of Purchases</b>	1,405,571,480.90
<b>Cost of Conversions</b>	7,295,935.95
<b>Cost of Activities</b>	11,999,880.00
<b>Cost of Inventories</b>	9,019,200.00
<b>Cost of Outsourcing</b>	0
	760,795,111.19
<b><i>Net Profit</i></b>	<b>OK</b>

Figure 23: Grand Summary



Profit Statement JAN 1997	
Done	Time <input type="text" value="1"/>
1/3	Revenue from Sales 838,343,608.05
Next	Cost of Purchases 505,471,480.90
Previous	Cost of Conversions 0.00
	Cost of Activities 0.00
	Cost of Inventory 6,400.00
	Cost of Outsourcing 0.00
	<i>Net Profit</i> 332,865,727.15

Figure 24: Profit Statement of One Time Period

### Interest Rate Data Entry

**Please enter the interest rate between two time unit in decimal number; for example if the interest rate between two months is 1.5% enter 0.015**

**Interest Rate**

Figure 25: Interest Rate Data Entry

Discounted Cash Flow		JAN 1997
<b>Done</b>	Time	<input type="text" value="1"/>
1/3		
<b>Next</b>	<b>Revenue from Sales</b>	817,896,202.97
<b>Previous</b>	<b>Cost of Purchases</b>	493,142,908.19
	<b>Cost of Conversions</b>	2,770,935.95
	<b>Cost of Activities</b>	0.00
	<b>Cost of Inventory</b>	6,243.90
	<b>Cost of Outsourcing</b>	0.00
	<b>Net Profit</b>	321,976,114.92

Figure 26: Discounted Cash flow

Discounted Grand Summary	
<b>Revenue from Sales</b>	2,084,107,383.89
<b>Cost of Purchases</b>	1,349,871,046.20
<b>Cost of Conversions</b>	0.00
<b>Cost of Activities</b>	11,421,658.53
<b>Cost of Inventory</b>	8,584,608.11
<b>Cost of Outsourcing</b>	0.00
<b>Net Profit</b>	714,230,071.03
	<b>OK</b>

Figure 27: Discounted Grand Summary

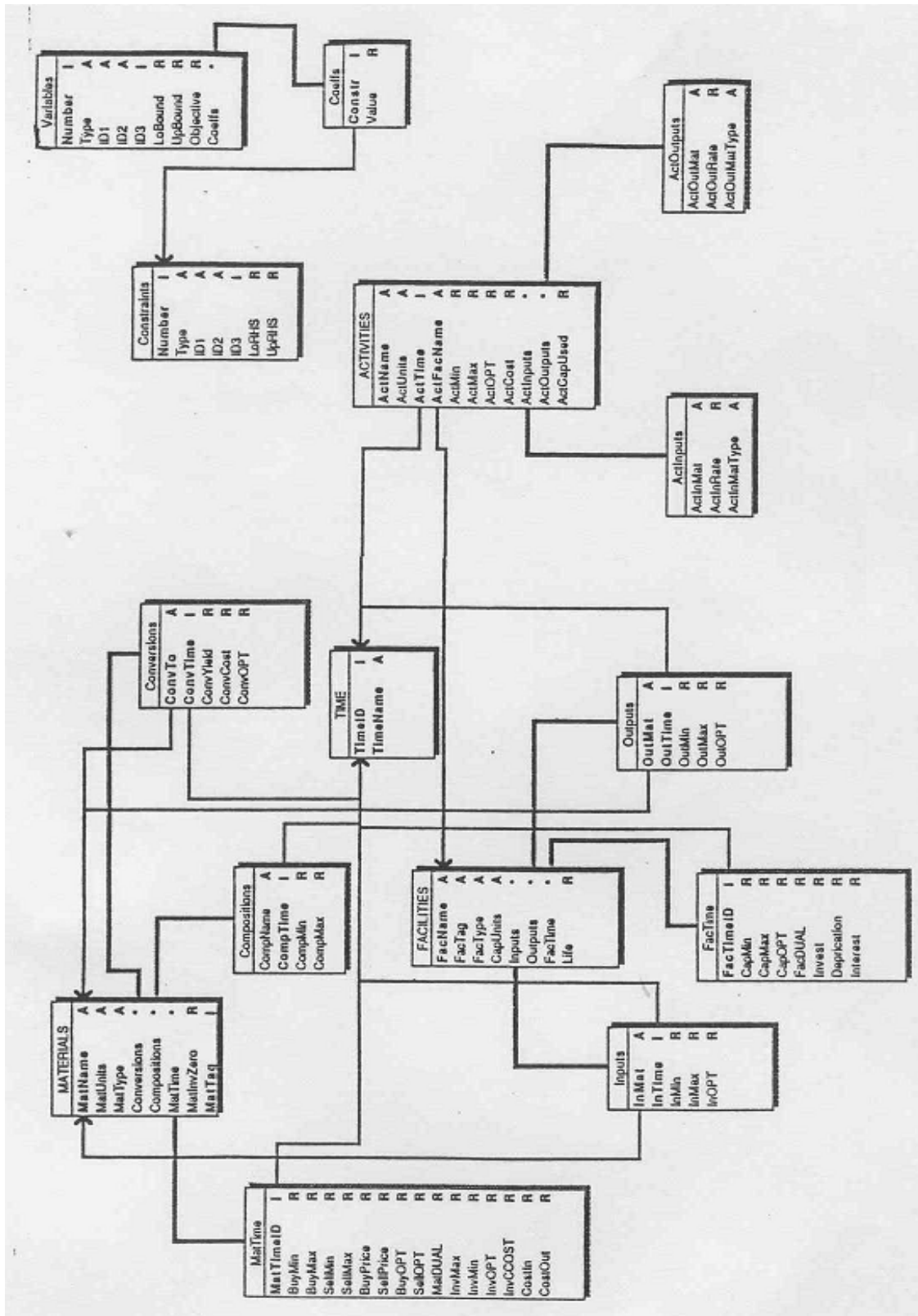


Figure 28: Database Structure of STEEL-TIME1

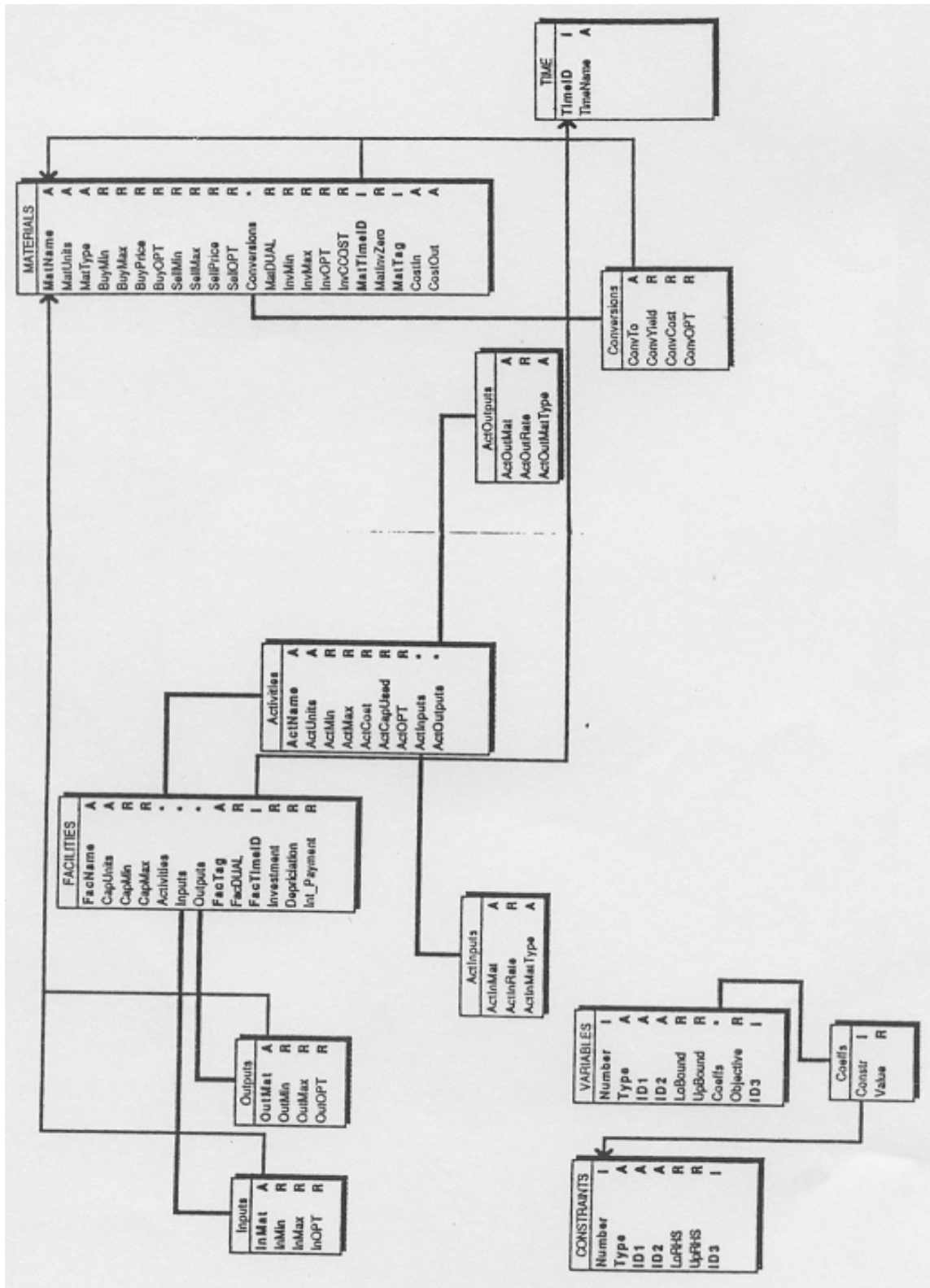


Figure 29: Database Structure of STEEL-TIME2