AIMMS

A Tutorial for Excel Users

April 2009

Paragon Decision Technology

Guido Diepen

Paragon Decision Technology B.V.
Schipholweg 1
2034 LS Haarlem
The Netherlands
Tel.: +31 23 5511512
Fax: +31 23 5511517

Paragon Decision Technology Inc.
5400 Carillon Point
Kirkland, WA 98033
USA
Tel.: +1 425 576 4060
Fax: +1 425 576 4061

Paragon Decision Technology Pte. Ltd.
80 Raffles Place
UOB Plaza 1, Level 36-01
Singapore 048624
Tel.: +65 96404182

Email: info@aimms.com
WWW: www.aimms.com

This documentation was typeset by Paragon Decision Technology B.V. using LATEX and the LUCIDA font family.

AIMMS tutorial for Excel users

Intended audience

This tutorial is intended for people who know how to use the solver in Excel and are considering to switch to AIMMS. It shows you what the AIMMS counterparts are when you are familiar with modeling in Excel. At the end of this tutorial you will have seen an implementation of a model in AIMMS, based on how it is modeled in Excel. This will enable you to start building your own AIMMS models.

Prerequisites

Install AIMMS

This tutorial assumes you have already installed AIMMS. If you have not done so yet, you can download the latest version of AIMMS and its prerequisites at http://www.aimms.com/latestdownload

Overview of the tutorial

Components

This tutorial consists of the following three components:

- This PDF file.
- An Excel file "AIMMS tutorial for Excel users - Excel model.xls". This Excel file contains two sheets: Input and Model. This separation has the following two reasons:
  - To make it easier to import data from the Excel file into AIMMS.
  - To make it easier to compare the Excel way of modeling with the AIMMS way of modeling.

  Throughout this tutorial "the Excel file" will refer to this Excel file.
- An AIMMS project in the "AIMMS tutorial for Excel users - AIMMS Model" subfolder. This AIMMS project consists of a number of files, of which the "AIMMS tutorial for Excel users - AIMMS Model.prj" is the main AIMMS file, with an AIMMS icon. If you have AIMMS installed, double-clicking on this project file should open AIMMS. Throughout this tutorial "the AIMMS project" will refer to this AIMMS project.

Example model

The model that is used as an example in this tutorial is based on one of the exercises in the book "Modeling the Supply Chain" by Jeremy F. Shapiro. This exercise is a purchasing optimization problem, where a computer manufacturer must buy a given number of customized hard drives. There are three different vendors, each having its own cost characteristics. Vendors may require a fixed cost for making the hard drives. Furthermore, the vendors use a tiered pricing scheme based on a certain price per disk for each hard disk ordered up to a given breakpoint, and a second, lower, price for each additional hard drive ordered above this breakpoint.

Overview remainder

The remainder of this tutorial is organized as follows: The first section shows how you can put the actual model in AIMMS, using the way it is modeled in Excel as a guideline. The second section explains how you can supply your model with data. The subsequent section shows how you can present the information of your model (both input and output) to an end-user. The final section offers tips and pointers on building your own models in AIMMS.

The optimization model

Outline of the section

In this section we will show what the AIMMS counterparts are for the information that is needed to use the solver in Excel. In Excel you provide the model details via the Solver Parameters window, an example of which can be seen in Figure 1.

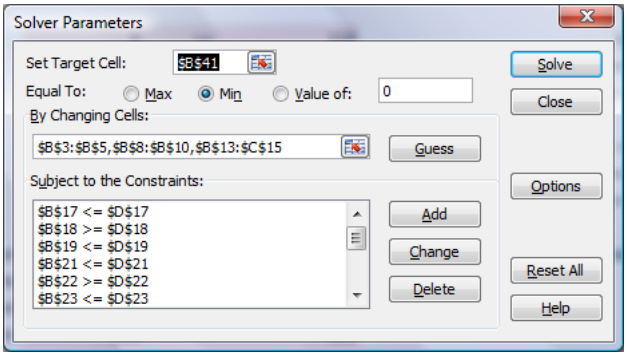

Figure 1: The Excel Solver Parameters window

Target Cell

In Figure 1, the target cell **B41** contains the total cost (i.e. the so-called objective) for a given purchase strategy. By selecting `Min`, it is denoted that we want to minimize the total cost. Furthermore, the cells that the solver can change (i.e. the so-called decision variables) and the constraints are also given. Finally, in the options "assume linearity" must be checked because in this tutorial we are working with linear models.

Identifiers

The elements of an optimization model (e.g. objective, variables, constraints, parameters) are represented in AIMMS by identifiers. These identifiers are structured in a tree-like manner within an AIMMS project. After you have opened a project in AIMMS, you can show this tree by pressing **F8**. The model tree corresponding to the example project in this tutorial is shown in Figure 2.

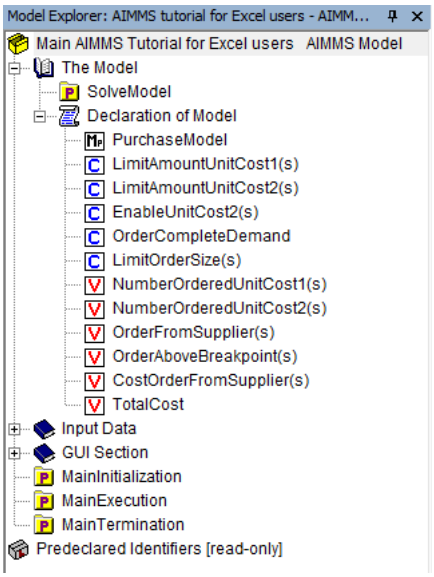

Figure 2: The AIMMS model explorer

Project structure

Figure 2 shows that the whole project has the following structure:

- Section "The model". This section contains all constraints, all variables, the model, and the procedure for solving the model.
- Section "Input data". This section contains all identifiers related to the Excel file and the input information.
- Section "GUI Section". This section contains all identifiers that are related to extra functionality provided by the Graphical User Interface (GUI).

Keeping the overview

Although it is possible to store all identifiers in the project without any structure, storing the model in such a structured way is one of the key advantages of AIMMS. It allows you to easily keep an overview of your model, even when it gets very large.

Attribute form

By double-clicking on an identifier, you will open up the attribute form of that particular identifier. Certain identifiers can have a tree of identifiers under them (e.g. the **Declaration** identifier "Declaration of Model" in the picture and the **Section** identifier "The Model"). For such identifiers there is a difference between double-clicking on the icon of the identifier and double-clicking on the name of the identifier:

- Double-clicking on the icon of the identifier will toggle between collapsing and uncollapsing of the tree under this identifier
- Double-clicking on the name of the identifier will open up the attribute form of the identifier.

Comments

To provide more background information, all of the identifiers used in the AIMMS model contain additional comments. You can double-click on any of the identifiers in the AIMMS model and you will see these additional comments in its attribute form. It is good practice to use this comment feature in your models to keep them maintainable.

Overview of counterparts

The following is a brief discussion giving the AIMMS counterpart for each of the steps of a model in Excel. The input data is represented by means of **Set** and **Parameter** identifiers.

Set

The different vendors Acme, Best, and Champion make up a set and this is represented in AIMMS by the set `Suppliers`.

Parameters

Each supplier has a number of parameters associated with it, such as the number of disks it can supply. For this parameter, a **Parameter** `MaximumAvailable` is used. An important aspect of this identifier is that it is indexed over the set `Suppliers`. This means that this parameter will have a value for each element in the set `Suppliers`, so one value for each supplier. Note that this is similar to how such parameters are written in symbolic notation: M_s denotes the maximum number of drives that supplier s can supply. There are also parameters that are not indexed. An example of this is the parameter `NumberOfDisksNeeded`. This is a so-called **scalar parameter** that only holds one value. How to fill these sets and parameters with data is discussed in the Section Data input / output.

Objective

The Target Cell **B41** in the Excel file contains the total cost of a purchase strategy. In the Excel file, this cell is defined to be the sum of the cost of each separate vendor. In AIMMS the total cost of a purchase strategy (i.e. the objective) is represented by the **Variable** `TotalCost`. By double-clicking on the `TotalCost` identifier, you can see that the `TotalCost` is defined to be the sum over the separate vendor costs.

Decision variables

The decision variables in the Excel file, specified in the field "By Changing Cells" tell the solver the values of which cells can be changed to optimize the value of the target cell. In AIMMS, decision variables are represented by **Variables**. In the Excel file, one part of the cells that can be changed is the range **B3:B5**. These three cells model the decision variable for each of the vendors whether hard drives are bought from that vendor or not, by taking the value 1 and 0 respectively. For this particular decision variable, the identifier `OrderFromSupplier` is used, which is indexed over the set `Suppliers`.

Intermediate values

If you want to calculate intermediate values (such as the total cost per supplier, given by the range **B37:B39** in the Excel file), you can do so by introducing variables. In the specific case of the cost per supplier, a **Variable** `CostOrderFromSupplier` has been introduced, which is indexed over the set `Suppliers`.

Constraints

The AIMMS counterpart of the Constraints in Excel are the **Constraints**. The three constraints **B17** <= **D17**, **B21** <= **D21**, and **B25** <= **D25** give an upper bound on the number of drives bought at Unit Cost 1 from vendor Acme, Best, and Champion respectively. In AIMMS, these three constraints are given by the **Constraint** `NumberOrderedUnitCost1`. This constraint is indexed over the set `Suppliers`, similar to the identifier `MaximumAvailable`.

Integer variables

Some of the constraints in Excel deal with defining certain variables as being integer or binary. In AIMMS, such typing is not done via constraints, but by changing the **Range** attribute of the **Variable**. In this attribute you can specify both an upper and lower bound, as well as the extra requirement that the value of the variable should be an integer.

Mathematical Model

Finally, the mathematical model itself is also represented with an identifier in AIMMS. While in Excel you have one mathematical model per sheet, in AIMMS you have to add a **Mathematical Program** to your model. In the AIMMS model this is the `PurchaseModel` identifier. The attribute form for this identifier is shown in Figure 3.

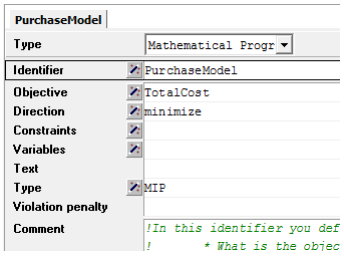

Figure 3: Attribute form for the **Mathematical Program** identifier

Solving the model

In Excel, you can start the actual solving of the model by pressing the **Solve** button on the Solver Parameters window. In AIMMS, you start the solving of `PurchaseModel` with the following statement in a **Procedure**:

```
solve PurchaseModel ;
```

The above code is put in the `SolveModel` procedure.

Attributes

The attributes of this identifier have clear similarities to the Solver Parameters window of Excel. By putting the `TotalCost` identifier in the Objective attribute and setting the direction to minimize, the solver is told what the goal is. Since the model contains both integer and binary variables, the **Type** attribute is set to MIP. You can also leave the **Type** attribute empty, in which case AIMMS will detect the model type automatically.

Default values

Finally, by leaving the **Constraints** and **Variables** attributes empty, the default value for these attributes is used. These defaults are the set of all constraints and all variables that are present in the project. In case you only have one mathematical program in your project, you can use these default values. When you want to have multiple mathematical programs, you have to provide each of the mathematical programs with its own set of variables and constraints.

Data input / output

Input sources

In the AIMMS project of this tutorial, all data input comes from the accompanying Excel file. This is just one of the possibilities to supply your model with data. Other possibilities include reading from text files, reading from XML files, and reading from ODBC and OLEDB data sources. Furthermore, the output of a model can be exported to these sources in a similar way. In this tutorial, these possibilities are not discussed, but further information about this can be found in other AIMMS documentation (e.g. the Language Reference and the User's Guide, both of which can be found in the **Help** menu of AIMMS).

Direct input

In case you are dealing with a very small model, you can also choose not to input the data from a separate source, but you can supply the data to your model directly within AIMMS. You can do this by opening up the attribute window by double-clicking on the identifier for which you want to enter the data. After that, in the attribute window you press the button (or alternatively you can press **CTRL-D**) to get the data input window. In Figure 4 the data entry window for a Set identifier is shown.

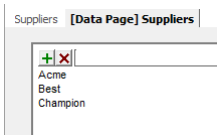

Figure 4: Data Page for a set identifier

Entering input

By entering the data in the textbox and pressing the + button in front of it, you can add new elements to the set. With the ✕ button you can remove a selected element from the set. Similar data input windows allow you to enter the data for scalar parameters and parameters that are indexed over a set.

Input from Excel

To ease the supply of data from the Excel file to AIMMS, named ranges have been created in the Excel file. Each of these named ranges corresponds to a separate part of the input (e.g. all suppliers). A screenshot of this Excel name manager is shown in Figure 5.

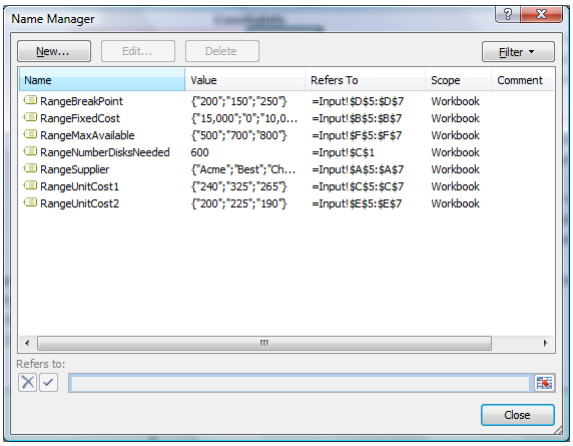

Figure 5: The Excel name manager

Named ranges

These names can be used to make it easier to refer to ranges of cells from within AIMMS. Not only are these names easy for referring to the cells from within AIMMS, also from within Excel you can use these names to refer to cells. For example, these named ranges can be used in the solver window to denote what the objective cell is, what the range of variable cells is, and for the constraints.

Procedure body

By double-clicking on the text of the `ReadDataFromExcel` procedure in the model tree, you can view the contents of the procedure. A screenshot of these contents is provided in Figure 6.

Additional comments in procedure body

The comment lines (all lines preceded by a '!' character are treated as comment lines in AIMMS) in the body of this procedure give a clear description of the AIMMS Excel functions `ExcelRetrieveParameter` and `ExcelRetrieveTable` needed for reading the data from the Excel file into AIMMS. Both approaches for referring to given cell ranges are used to show the exact use of named ranges and direct cell references.

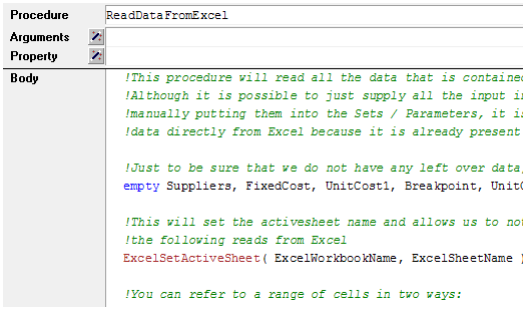

Figure 6: Attribute window of the `ReadDataFromExcel` procedure

Presenting information to the end-user

Building GUI: Pages

One key advantage of AIMMS is that it is easy to build a GUI to enable an end-user to work with your model: to supply the model with data, to select certain options for the model, to run a solve, and to analyze the results after the solving is finished. The concept used for this is "pages". Building pages is as easy as dragging an object on a page and specifying the attributes of this object.

Templates

You can use templates to make the pages of the project consistent. A second version of the AIMMS project used in this tutorial is present in the examples that come with the AIMMS installation and has the name `Purchase Optimization`. This version uses the standard AIMMS example template style. If you look at all examples supplied with AIMMS, you will see that they all have the same global layout. This is all done via templates.

Page tree

Similar to the identifiers in the model tree, the pages are ordered in a page tree. You can show this page tree using the **Page Manager** by either pressing the  icon in the toolbar, or by pressing the **F9** key. Since each page can hold a sub-tree of pages underneath it, the same actions for double-clicking on the name or on the icon of a page hold as for double-clicking on identifiers that can hold a sub-tree: double-clicking on the icon uncollapses the sub-tree, while double-clicking on the actual name of the page opens the page.

View/Edit mode

One of the major points to keep in mind is that for the pages three different modes are available, the two most important of which are: page view mode and page edit mode. In view mode, you are actually using the page as an end-user, while in edit mode, you can edit the layout of the page as a developer. You can toggle between these two modes by either pressing the **F4** key, or by pressing the  and icon for going to edit and view mode respectively. When you have a page in edit mode, you can add new objects to this page (either by

clicking on the shortcuts in the toolbar, or via the **Object** menu).

Resize edit mode

The third mode is the Page resize edit mode. This mode can only be selected via the View menu and it helps you to determine how pages behave when they are resized. This way a consistent look can be obtained for different screen resolutions.

Property window

Double-clicking on the page or any of the objects on a page opens the properties window for the page or object respectively. If you double-click in edit mode on the chart in which the optimal purchase strategy is displayed, you will see the window as shown in Figure 7, in which the different properties for this 2D Chart are presented.

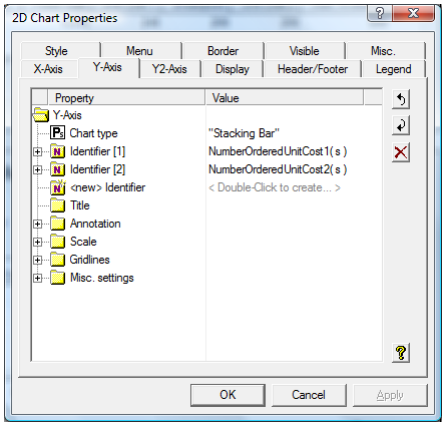

Figure 7: Property window for the 2D Chart object

Chart properties

On the X-Axis tab of this chart object, we have specified that the domain for the chart is the index s of the set `Suppliers`. Figure 7 shows the Y-Axis tab, on which you can see that the type has been set to *Stacking Bar* and that per supplier s on the X-Axis (i.e. the domain), we plot the value of the number of disks ordered at unit cost 1 and unit cost 2.

Further information

More detailed tutorials on how to create the GUI pages can be found in the Tutorial for Beginners and the Tutorial for Professionals, both of which can be found in the **Help** menu.

Creating your own AIMMS project

New project

When you start AIMMS, you can create a new project by either clicking "Create a new Project" on the start page, or via "New Project" in the **File** menu. When you create a new AIMMS project, the model tree will be in the following default state:

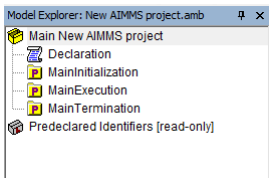

Figure 8: Default model explorer for a new AIMMS project

Sections

You can divide your complete project into different sections, by adding a section identifier for each section to the model tree. By double-clicking on the icon (not on the text) of the section, you can display the sub-tree and start adding identifiers to the sub-tree of the section.

Extend model

You can add identifiers to the model tree in the following way:

- Click with the mouse on the place where you want to insert the identifier
- Click on the ... icon in the menu bar or press the Insert key to retrieve a list of possible identifiers that you can insert at that point in the model tree. Please note that most of the identifiers need to be under a Declaration identifier. Some exceptions to this are Declaration itself, Section, and Procedure.
- Select the type of identifier you want to add to the model tree
- Type in the name you want to use for this identifier

Attribute window

Recall that you could show the attribute window for an identifier by double-clicking on the name of the identifier and that you can unfold the sub-tree under certain identifiers (e.g. Declaration and Section) by double-clicking on the icon of the identifier.

Wizard

When you have opened the attribute window, certain attributes can be specified by using a wizard. By pressing the button before the input field, or selecting the input field and pressing **CTRL-W**, you will see a wizard dialog box that helps you to specify the value for this attribute.

Further help

In case you run into problems, you can press **F1** to show the help window. Additionally, in the **Help** menu, all tutorials and manuals are accessible as PDF files. In case the help and these additional documents do not provide an answer to your question, you can post your question at the AIMMS Google group at http://groups.google.com/group/aimms.

Finally, we hope you will enjoy building your own optimization models in AIMMS!