

Tutorial for G-AIMMS: Witch Apprentice - Spell Assignment

Introduction to G-AIMMS

G-AIMMS is a series of AIMMS based games aimed to extend your modeling skills and AIMMS knowledge. G-AIMMS: Witch Apprentice - Spell Assignment is the second level of Witch Apprentice. New AIMMS users are recommended to read this tutorial in case they have problems finishing the level. You can always review this tutorial while playing G-AIMMS. It is located on the Help page.

Solving a problem in AIMMS

To solve a problem in AIMMS, you only need to worry about formulating a good model. Finding the solution to your model is easy using one of the solvers included with AIMMS. AIMMS clearly differentiates between the model and the data. In this level all the data you need is provided to you at the start of the level via predeclared identifiers located in the Input Data section of the model. Your task is to formulate a good model. All basic information about formulating a model, such as the use of the Model Explorer and the different type of relevant identifiers is explained in the tutorial of the first level, Gathering Ingredients. Have a look at that tutorial in case you want to read about these general tips again.

Index Domain Conditions

In the previous tutorial we explained the parameter $\text{Distance}(i,j)$ with index domain $(i,j)|i \neq j$ that represents the distance between two cities i and j . This index domain makes sure the parameter is defined for all pairs of cities (i,j) under the condition that i is unequal to j . When we use an index domain condition for a variable or constraint, this directly influences the generated mathematical program. In case we have a parameter $\text{AllowedTransport}(i,j)$ which contains 0/1 values indicating between which locations transport is allowed and we have a variable $\text{Transport}(i,j)$ we can restrict the solver to only allow the variable Transport to get values for those (i,j) for which the transport is allowed by giving it the index domain: $(i,j) | \text{AllowedTransport}(i,j)$

Element Parameters

In the input data of this level we have two element parameters, SpellCategory and WitchRegion . An element parameter is a parameter with as value an element instead of a numerical value. In our model $\text{SpellCategory}(s)$ has as value for each spell s a category, and $\text{WitchRegion}(w)$ has a value for each witch a region. We use these element parameters to indicate to which category a spell belongs and in which region of the country a witch lives. An element parameter has a range attribute in its declaration form that should contain the range set. The element parameter can only get values that are elements from the range set.

The type of relationships that can be established with element parameters can be useful in certain type of restrictions. Let's assume we have sets of Cities (index c) and States (index s) in the USA, and an element parameter CityState indicating the state in which a city is located, and a decision variable $\text{PlaceDistributionCenter}(c)$ to decide whether we place a distribution center in a certain city. If we have a limit on the number of distribution centers in a state, $\text{MaxDistributionCentersInState}$, we can restrict our model by introducing the following constraint:
 $\text{RestrictMaxDistributionCenters}(s): \text{Definition: } \sum(c | \text{CityState}(c) = s, \text{PlaceDistributionCenter}(c)) \leq \text{MaxDistributionCentersInState}.$

Playing this G-AIMMS

Below is an overview of the steps you have to take to complete this G-AIMMS:

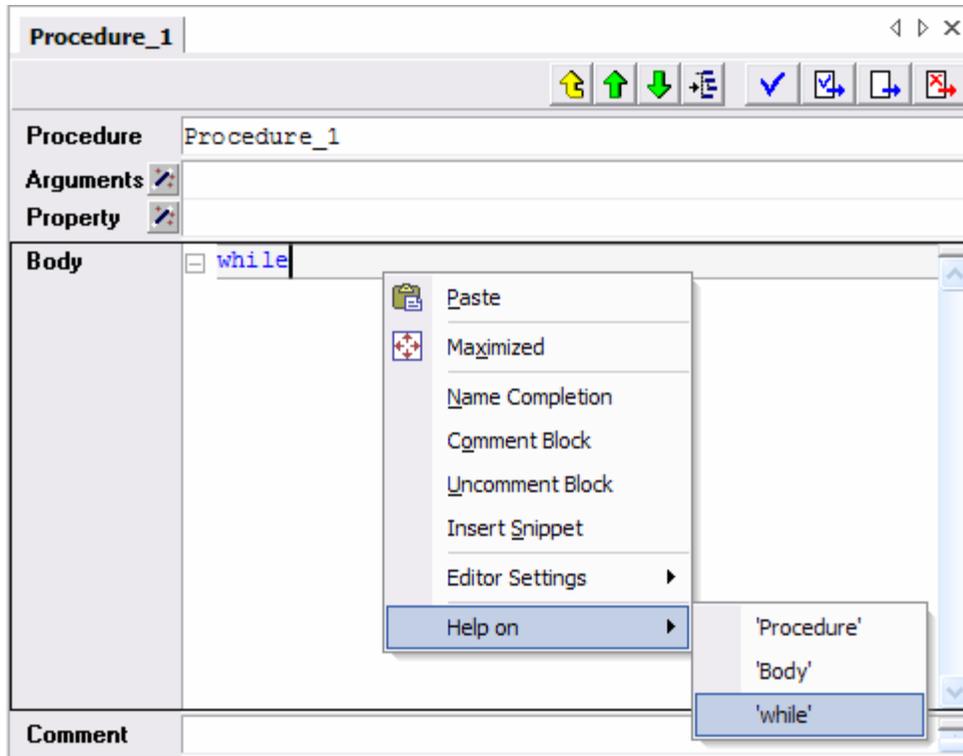
- Open the file 'SpellAssignment.prj'. This will start this G-AIMMS. For this G-AIMMS you can use your own AIMMS license, a free trial license, or a free student license. You need a password to open this project, the password can be acquired by completing the previous level.

- Take a look at the story, the problem description and the input data to get an impression of the problem.
- Open the Model Explorer. Take a look at all identifiers declared in the Declaration section located in the Input Data section. You can use all these identifiers in your model, for example in the definition of constraints. All data for these identifiers is already provided, as you can see when you open the data pages of these identifiers.
- Now the time has come to build your model. You can declare your identifiers and procedures in the Player Section of the Model Explorer. Remember that identifiers always have to be declared in a Declaration section, while procedures have to be declared outside of such a section. For this level you will only need to declare variables, constraints, and a mathematical program. A procedure with the name SolveSpellAssignmentProblem has been predeclared for you in the Player Section. If you open the attribute page of this procedure, you will see comment text describing the steps you have to take and some more information.
- Run the procedure that solves your model. You can press CTRL+p to open the Progress Window to see the progress that has been made solving your model. When the solver is done, all of your declared variables have gotten their optimal values. You can open the data pages of the variables to see those values. Should the solver take a long time to solve your model you probably either made a mistake in your model or you did not formulate your model efficiently enough. In this case you can press CTRL+SHIFT+s to abort the solver and change your model.
- The easiest way to check your solution is to first assign your solution to the parameter sl::SuggestedWitchSpellAssignment(w,s). If you go to the Submit Solution page your solution should be displayed in the table. Alternatively, you can input your solution values in this table by hand. You can now press the 'Check Solution' button to check your solution. Should your solution be incorrect, G-AIMMS will tell you why it is incorrect. In this case, please correct your model. If your solution is correct, congratulations on solving this G-AIMMS: Witch Apprentice level!

Where to find help while playing

While playing the G-AIMMS, there are a lot of ways to get help. See below for some of the options:

- You can always review this tutorial. It is located on the Help page.
- You can look for extra help in the Help menu, where you can find links to the complete AIMMS documentation and where you can search all of the documentation for a certain topic.
- You can click the  icon and select an object on your screen to get help on that object.
- You can type the word you want help on in a procedure, press the right mouse button while your cursor is above it and select 'Help on'.



Final Tips

Below you can find some final tips before you start playing this level:

- Should you accidentally close all of the pages, you can open the start page by clicking the Open first page icon  at the top of your screen. You can also use the File menu to do so.
- Remember to often save your project. You can do so by pressing the Save all icon  at the top of your screen or by using the File menu.
- Try to keep all constraints in your model linear, which means you should avoid products of variables in constraints. Products of parameters and variables are fine. If your model is nonlinear, it is much harder to solve and solving it may lead to inexact solutions.
- If you feel like you need more knowledge on AIMMS to play this level, you can do the tutorial for beginners first. It is linked in the Help menu in G-AIMMS under 'Additional Documentation'.
- The best way to learn how to use AIMMS, is to just try it, so go ahead and play G-AIMMS!