**Background**

Engineering Optimization Capabilities and Methods:

The Engineering Optimizer has the ability to take a parameter and drive it toward a defined optimized value by varying other free parameters. First, a parameter needs to be chosen that will be the optimized parameter. Then an optimizing type must be chosen. Optimizing types are as follows: Maximize (find the biggest value), Minimize (find the smallest value), Target Value (find a value closest to a desired value), and Only Constraints (try to move toward values that satisfy all constraints). The parameters that can be optimized are the dimensions of the part, as well as functions defined using CATIA’s Formula Editor. Next, all of the free parameters need to be defined. These generally are picked from the dimensions that make up the part, or functions using those dimensions. Last, but not least, constraints need to be defined to insure that the optimizer's solution is feasible.

When all of this is done correctly the optimization can be run using the one of the five optimization algorithms. The optimizer fixes all but one of the variables and varies that remaining variable, looking for a trend that moves toward an optimized value. After some number of iterations, the optimizer changes the variable to be varied and continues with the process. It does this for a certain number of total iterations (defaulted at 200), or for a certain number of iterations without improvement toward the goal (defaulted at 50), or a maximum amount of time taken on the optimization (defaulted at 5min). The Optimizer will take the best permutation of the free variables and apply them to the part, and can be shown on the part if visualization is enables. The Optimizer can also save the iteration data to an Excel Spreadsheet file that will help to track the Optimization process as shown in this figure.

**Engineering Optimizer**

Formula Formation:

 Using the formula editor in the Knowledge Advisor workbench, formulas can be entered to define aspects of a part or assembly as a function of existing measurements or features. These equations can be entered manually in the editor, or built using functions and elements in the editor’s dictionary, including part measurements such as smartvolume, and logical operators. Features or parameters can be entered into the equation by double-clicking them in the design tree. Once entered, a formula-derived parameter can be used in the optimizer.

Variables:

 The Engineering Optimizer can use parameters that are in the part as either free parameters or the optimized parameter. All of these variables start out as a predefined generic name assigned by CATIA, such as Part/Body1/line3. In order to make them easy to access during optimization it is recommended to rename some of these variables to match their respective relations. Instead of looking for Part/Body1/line3, it is much easier to find Height. To do this right click on a dimension and then select the dimensions name from the drop down menu, and finally select rename parameter. In this way we defined the Total Height of our beam as shown.

Constraints:

 Constraints are a way for the experimenter to guide the optimization. They tell CATIA the design limitations that are on a part or a feature. Constraints are put on an existing parameter, which can be user defined or found through any of the provided filters that are built into CATIA. In the Product Engineering Optimizer you can use logic operators to set the range of a parameter. The Optimizer Constraint equation needs a number on the right side in order to work. To set one variable equal to another all that has to be done is to set the difference of the variables equal to zero, such as Variable1-Varible2==0, which is just an algebraic manipulation of the equation Variable1==Variable2. There is also possibility to set a weight to a constraint to make it more valuable than others. This is useful when the problem has many constraints, or a varying level of importance between one constraint and another.

**Our Experiment**

Problem Statement:

 Suppose a structural support is needed for a shelf or platform. The design calls for a 1-meter long I-beam to be constructed out of sheet metal, but there is a weight limit on the beam. What should the height and width of the beam’s cross-section be, and what thickness of sheet metal should be used in order to minimize any bending in the final structure while staying within the weight limit?

Setup of I-beam:

 An I-beam was created by first drawing a cross-section and extruding it one meter. One quarter of the beam shape was drawn, and the profile was then mirrored in order to ensure a symmetric shape about the y- and z-axis that was centered on the origin. Then parameters that were necessary in the optimization were renamed to make them easier to find (Width being the width of the base, Height being the total height of the beam, Web being the thickness of the web, and Flange being the thickness of the flange).

Defining our Parameters:

 We used two functions in our I-beam experiment. The first was smartvolume. This was a simple matter of finding smartvolume in the function library and applying it to the I-beam part body. The second was Inertia. We derived the Inertia equation for this optimization from scratch, using parameters that we defined when we sketched the I-beams cross-section (Width being the width of the base, Height being the total height of the beam, Web being the thickness of the web, and Flange being the thickness of the flange). The derivation of this function is shown on the following page.

Since Inertia can be found in parts we divided our cross-section into two parts. The first part is the inertia of a rectangular cross-section of the total width and height values about the x axis:

$$\frac{Width×(Height)^{3}}{12}$$

And the second was the two rectangular gaps that would subtract from the total Inertia of the beam:

$$-\frac{(Width-Web)×(Height-2×Flange)^{3}}{12}$$

So the total Inertia is the Sum of the partial Inertias:

$Ixx=\frac{Width ×(Height)^{3}}{12}-\frac{(Width-Web)×(Height-2×Flange)^{3}}{12}$

**Our Optimization:**

Setting up the Problem:

 To make this problem tie to geometric features of the beam so that we could solve it in CATIA we said that if we maximize the moment of inertia about the x axis then we would be minimizing the deflection of the beam. Therefore our selected optimization type is maximization and our optimized parameter is the function that we derived for *Ixx*. The parameters that we chose to allow CATIA to vary are the Height, Width, Web, and Flange. We let CATIA automatically chose the step size to vary these with, but if the experiment required a certain step on any of the parameters it is a simple matter of right clicking on the parameter to define the Inferior Range, Superior Range, and Step Size. Also we chose the Simulated Annealing Algorithm for the algorithm type. We left the Convergence speed at fast. We also left the Termination Criteria at their default settings. By checking the box next to “Save Optimization Data” we told CATIA to create a spreadsheet in excel of all of its iterations. By choosing the with visualization update we told CATIA to apply the dimensions that it determined were the best to the part geometry in the visualization window. We disconnected the undo log so that we wouldn’t fill it up with 200 iterations of this part. All this is shown in the figure.

Our Constraints:

We set the optimizer to hold the beam’s volume at a specific value (.01m3), to simulate the weight limit in the problem. Since this whole beam was supposed to be constructed from one type of sheet metal, formulas were also entered to set the flange thickness and web thickness equal to each other. Finally, in order to preserve an actual I-beam shape, the total width was constrained to keep it from being less than the web thickness, the height was set to be greater than or equal to twice the flange thickness, and the width, web, and height were set to be greater than zero. Since the constraints require one side of the equation to be a numerical value, some of these constraints had to be rearranged. Take note that in the Simulated Annealing Algorithm the "==" constraint will never be satisfied due to how it runs its analysis, but in the distance to satisfaction column the number displayed is essentially zero.

Computation Results:

 Once our optimization was complete, it produced an excel document as explained above and resized our model in the visualization window as shown. As expected, CATIA made the part taller to pull more of the mass away from the x axis. In doing so it increased the moment of Inertia.

**Possible Future Experiments:**

* Experimentation with the “only constraints” optimization in order to optimize a part based on weighted constraints.
* Cost analysis of optimizations, either in CATIA, or with the excel spreadsheet generated by the optimizer.
* Optimizing a design for ease of construction, such as minimizing required welding or machining.
* Using the optimizer to improve the design of assemblies containing multiple parts.
* Including constraints to allow for tool and fastener clearances.