**Overview – Julia, JuMP, and Solvers**

In this class, you will be using Julia, JuMP, and different solvers to solve optimization problems. Here is a breakdown of each of these components.

|  |
| --- |
| **Julia**  Julia is a high-performance dynamic **programming language** for technical computing developed by MIT students. It has several **packages** that can be added. |

|  |
| --- |
| **JuMP**  JuMP is a **package** for Julia that we will be using for optimization problems. It is a language for mathematical programming problems.  JuMP allows you to:   * Create a model * Define decision variables and constraints * Set an objective, etc. |

|  |
| --- |
| **Solvers**  JuMP supports a number of open-source and commercial **solvers** including COIN Clp, COIN Cbc, Gurobi, etc.  The **solver** is what solves the optimization models.  Solvers can solve   * Linear optimization problems * Mixed integer problems * Nonlinear optimization problems, etc.   In this class, we will either use JuMP's built in solvers (COIN Clp/Cbc), or Gurobi which you can download from online. Please see the ***"Gurobi Installation Guide"*** to download Gurobi. |