

- Throughout this course, we've focused on Julia's `Optim` package

- Throughout this course, we've focused on Julia's `Optim` package
- This package provides algorithms for nonlinear unconstrained optimization

- Throughout this course, we've focused on Julia's `Optim` package
- This package provides algorithms for nonlinear unconstrained optimization
- This makes sense: likelihood functions are nonlinear

- Throughout this course, we've focused on Julia's `Optim` package
- This package provides algorithms for nonlinear unconstrained optimization
- This makes sense: likelihood functions are nonlinear
- But there are many other types of optimizers out there

- Throughout this course, we've focused on Julia's `Optim` package
- This package provides algorithms for nonlinear unconstrained optimization
- This makes sense: likelihood functions are nonlinear
- But there are many other types of optimizers out there
- They may not be as applicable for econometric applications

- Throughout this course, we've focused on Julia's `Optim` package
- This package provides algorithms for nonlinear unconstrained optimization
- This makes sense: likelihood functions are nonlinear
- But there are many other types of optimizers out there
- They may not be as applicable for econometric applications
- But they might be helpful for certain applications

Aside from nonlinear optimization, there are:

Aside from nonlinear optimization, there are:

- Linear programming



Aside from nonlinear optimization, there are:

- Linear programming
- Mixed integer programming

Aside from nonlinear optimization, there are:

- Linear programming
- Mixed integer programming
- Semidefinite programming

Aside from nonlinear optimization, there are:

- Linear programming
- Mixed integer programming
- Semidefinite programming
- Convex optimization

Aside from nonlinear optimization, there are:

- Linear programming
- Mixed integer programming
- Semidefinite programming
- Convex optimization
- Constrained nonlinear optimization

Aside from nonlinear optimization, there are:

- Linear programming
- Mixed integer programming
- Semidefinite programming
- Convex optimization
- Constrained nonlinear optimization

Each of these algorithms can be accessed through Julia's JuMP package

These other algorithms in JuMP have valuable real-world uses:

These other algorithms in JuMP have valuable real-world uses:

- Optimal bus routes

These other algorithms in JuMP have valuable real-world uses:

- Optimal bus routes
- Optimal power grid architecture



These other algorithms in JuMP have valuable real-world uses:

- Optimal bus routes
- Optimal power grid architecture
- Solving budget constraint problems

These other algorithms in JuMP have valuable real-world uses:

- Optimal bus routes
- Optimal power grid architecture
- Solving budget constraint problems
- Solving sudoku puzzles

These other algorithms in JuMP have valuable real-world uses:

- Optimal bus routes
- Optimal power grid architecture
- Solving budget constraint problems
- Solving sudoku puzzles

The great thing about JuMP is that it [interfaces](#) with a plethora of optimizers

These other algorithms in JuMP have valuable real-world uses:

- Optimal bus routes
- Optimal power grid architecture
- Solving budget constraint problems
- Solving sudoku puzzles

The great thing about JuMP is that it **interfaces** with a plethora of optimizers

You can keep your code the same and simply switch out which optimizer to use