Teaching Optimization
[JuMP, cvxpy, Pyomo]

Bernardo Freitas Paulo da Costa (UFRJ & FGV)

July 19 2024
JuMP-dev, Montréal

UFRJ

FGV EMAp

A bit of history

- Background in pure math (up to PhD), with a good deal of Computer Science;

A bit of history

- Background in pure math (up to PhD), with a good deal of Computer Science;
- Started research in (Stochastic) Optimization in 2016;
- First course in convex optimization with `cvxpy`;

A bit of history

- Background in pure math (up to PhD), with a good deal of Computer Science;
- Started research in (Stochastic) Optimization in 2016;
- First course in convex optimization with `cvxpy`;
- Industry project started in 2017 using `SDDP.jl` (and `julia-0.6`);
- Developed helper packages (interfacing JuMP & solvers) in 2019, in the Julia-1.0 transition

A bit of history

- Background in pure math (up to PhD), with a good deal of Computer Science;
- Started research in (Stochastic) Optimization in 2016;
- First course in convex optimization with `cvxpy`;
- Industry project started in 2017 using `SDDP.jl` (and `julia-0.6`);
- Developed helper packages (interfacing JuMP & solvers) in 2019, in the Julia-1.0 transition
- More SP/`JuMP` packages since 2021 (`SamplingRB.jl`, `DualSDDP.jl`, `LDR.jl`)

A bit of history

- Background in pure math (up to PhD), with a good deal of Computer Science;
- Started research in (Stochastic) Optimization in 2016;
- First course in convex optimization with `cvxpy`;
- Industry project started in 2017 using `SDDP.jl` (and `julia-0.6`);
- Developed helper packages (interfacing JuMP & solvers) in 2019, in the Julia-1.0 transition
- More SP/`JuMP` packages since 2021 (`SamplingRB.jl`, `DualSDDP.jl`, `LDR.jl`)
- Two courses using `Pyomo` for MBA@UFRJ in Data Science (Zoom lectures)
- Short `JuMP` course in Summer Program @ FGV (inspired from Oscar's 2023 tutorials)

My experience teaching optimization

- All frameworks need a *math introduction* (including modeling)

My experience teaching optimization

- All frameworks need a *math introduction* (including modeling)
- It takes a “click” to relate “Models – Applications”

My experience teaching optimization

- All frameworks need a *math introduction* (including modeling)
- It takes a “click” to relate “Models – Applications” (and auxiliary variables)

My experience teaching optimization

- All frameworks need a *math introduction* (including modeling)
- It takes a “click” to relate “Models – Applications” (and auxiliary variables)
- Very close to mathematical notation;
- “Program variables” *vs* “Problem variables”;

My experience teaching optimization

- All frameworks need a *math introduction* (including modeling)
- It takes a "click" to relate "Models – Applications" (and auxiliary variables)
- Very close to mathematical notation;
- "Program variables" *vs* "Problem variables";
- Good (fast, reliable, free) solvers make a *big difference*!

What I like about cvxpy

What I like about `cvxpy` and probably `Convex.jl`

What I like about cvxpy and probably `Convex.jl`

- Object-Oriented seems more natural than macros:
    - `x = Variable()` *vs* `@variable(m, x)`;

What I like about cvxpy and probably `Convex.jl`

- Object-Oriented seems more natural than macros:
    - `x = Variable()` *vs* `@variable(m, x)`;
- *Convexity checks* is good for teaching convex optimization and Disciplined Convex Programming rules;

What I like about cvxpy and probably `Convex.jl`

- Object-Oriented seems more natural than macros:
  - `x = Variable()` *vs* `@variable(m, x)`;
- *Convexity checks* is good for teaching convex optimization and Disciplined Convex Programming rules;
- Enforces building a *computational graph*

What I like about JuMP

- Access to *many solvers*
- More model flexibility
  - Students see more applications and flexibility,

What I like about JuMP

- Access to *many solvers*
- More model flexibility
  - Students see more applications and flexibility,
  - Most projects students propose end up falling out of the convex setting;

What I like about JuMP

- Access to *many solvers*
- More model flexibility
  - Students see more applications and flexibility,
  - Most projects students propose end up falling out of the convex setting;
- Variables are attached to models, so we don't "pirate";

What I like about JuMP

- Access to *many solvers*
- More model flexibility
  - Students see more applications and flexibility,
  - Most projects students propose end up falling out of the convex setting;
- Variables are attached to models, so we don't "pirate";
- The documentation;

What I like about JuMP

- Access to *many solvers*
- More model flexibility
  - Students see more applications and flexibility,
  - Most projects students propose end up falling out of the convex setting;
- Variables are attached to models, so we don't "pirate";
- The documentation;
- Re-solves (and hoping for ever more efficient warm-starts).

My (limited) Pyomo experience

- Python is still more widespread among students;

My (limited) Pyomo experience

- Python is still more widespread among students;
- Can also use many solvers and is not restricted to convex problems;

My (limited) Pyomo experience

- Python is still more widespread among students;
- Can also use many solvers and is not restricted to convex problems;
- The `AbstractModel` is very close to how we think, and teach (and research);

My (limited) Pyomo experience

- Python is still more widespread among students;
- Can also use many solvers and is not restricted to convex problems;
- The AbstractModel is very close to how we think, and teach (and research); (in JuMP: see the Knapsack / design patterns tutorial)

My (limited) `Pyomo` experience

- Python is still more widespread among students;
- Can also use many solvers and is not restricted to convex problems;
- The `AbstractModel` is very close to how we think, and teach (and research); (in `JuMP`: see the Knapsack / design patterns tutorial)
    - Generator expressions are powerful, but some students find it too "magical".

Things I would like in JuMP

With apologies if they already exist. . .

Things I would like in JuMP

With apologies if they already exist...

- *Abstract Linear Algebra*, especially for structured matrices;

Things I would like in `JuMP`

With apologies if they already exist...

- *Abstract Linear Algebra*, especially for structured matrices;
- Convexity checking / `Convex.jl` integration;

Things I would like in JuMP

With apologies if they already exist. . .

- *Abstract Linear Algebra*, especially for structured matrices;
- Convexity checking / `Convex.jl` integration;
- Reduce TTFx without a 20 GB `.julia`.

- Comments?
- Suggestions?