Solving large-scale NLP problems with JuMP and Knitro

François PACAUD

June 25, 2019

Introduction

Who am I?

- Currently optimization engineer at Artelys
- Working on the development of Knitro

And Artelys?

- A french PME (about 70 employees)
- Specialized in the optimization of energy systems

What could you expect from this talk?

- No math!
- Short presentation of the solver Knitro
- Feedbacks & thoughts on developing a wrapper based on MOI
- Of course, a case study with a benchmark! :)

When Knitro met JuMP

Knitro?

$$\min_x \; f(x) \quad x_L \leq x \leq x_U \;, \quad c_L \leq c(x) \leq c_U$$

- Generic non-linear optimization (NLP) solver
- Based on interior-point algorithms
- Extension to Mixed Integer Non-linear optimization (MINLP) by using Branch & Bound
- Other interesting features:
  - Second order cone programming (SOCP)
  - Complementarity constraints
  - Non-linear least squares (and by extension non-linear equations)
  - Derivative free optimization (finite differences)

Cover a broad range of optimization problems!

Knitro timeline

- Development started in 2001 at Northwestern University
- Originated by Richard Waltz (former PhD student of Jorge Nocedal)
- Development first backed by Ziena Optimization
- 2015: Ziena bought by Artelys
- ...

Knitro timeline

- Development started in 2001 at Northwestern University
- Originated by Richard Waltz (former PhD student of Jorge Nocedal)
- Development first backed by Ziena Optimization
- 2015: Ziena bought by Artelys
- ...
- **2018: new incremental API for Knitro**
- **2018: major update in Knitro.jl!**

Recent development in 2018

Developments of MathOptInterface (MOI)

```
MOI.add_variable(model, ...)
```

Release of Knitro 11.0, with new procedural API

```
KN_add_var(kc)
```

Perfect match!

Specifying NLP models in generic modelers

Specifying gradients $\nabla f(x)$, Jacobians $\nabla c(x)$
and Hessians $\nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(x)$ to the solver?

- Rely heavily on automatic differentiation!
- Generic numerical problem, different solution (AMPL, Casadi)
- JuMP: relies on its own implementation of ForwardDiff.jl
  (graph coloring approach)

Feedbacks on implementing Knitro.jl

- Knitro.jl originally developed by JuliaOpt
  (credits to Yeesian Ng and Miles Lubin)
- Previous version of Knitro.jl worked with MathProgBase
- Porting Knitro.jl to MOI:
  - Incremental API efficiently wrapped with MOI
    (thanks to a well-studied design!)
  - But ...

Feedbacks on implementing Knitro.jl

- Knitro.jl originally developed by JuliaOpt
  (credits to Yeesian Ng and Miles Lubin)
- Previous version of Knitro.jl worked with MathProgBase
- Porting Knitro.jl to MOI:
  - Incremental API efficiently wrapped with MOI
    (thanks to a well-studied design!)
  - But ...
    - knitroampl: written in pure C on top of ASL

Feedbacks on implementing Knitro.jl

- Knitro.jl originally developed by JuliaOpt
  (credits to Yeesian Ng and Miles Lubin)
- Previous version of Knitro.jl worked with MathProgBase
- Porting Knitro.jl to MOI:
  - Incremental API efficiently wrapped with MOI
    (thanks to a well-studied design!)
  - But ...
    - knitroampl: written in pure C on top of ASL
    - MOI: issues with Julia's garbage collector (finalizer)
      Non-deterministic segfaults during development...

Leveraging JuliaOpt in Knitro's development

Knitro's development benefits from JuliaOpt community.

- Add new functions in Knitro's API to improve MOI support
- Add MINLPTests.jl to Knitro's continuous integration
- Integration of new benchmarks with MathOptFormat.jl

Use-cases

NLP? What for?

- Optimal control
- Chemical engineering (Grossman, Biegler, Waechter,...)
- Optimal power flow (OPF)
- Economics
- Machine learning (see [Bottou, Curtis, Nocedal 2018])

Focus here on OPF!

Introducing PowerModels.jl

Optimal Power Flow in a nutshell:

- Electrical engineering problem:
  - Given resistances, generations and demands...
  - ... find optimal dispatch of power flows in the electrical network
- Different ways to frame as an optimization problem
- Difficult NLP problems

PowerModels.jl allows to formulate OPF problems with JuMP

- Developed by Carleton Coffrin at Los Alamos National Laboratory
- Lot of flexibility!

Benchmarking Knitro with the PGLIB benchmark

Objective: solving the PGLIB benchmark with Knitro

- Large-scale problems (up to 80,000 variables)
  Suitable for a benchmark!
- Instances of PGLIB prove to be challenging

Running Knitro on the PGLIB benchmark, we observed

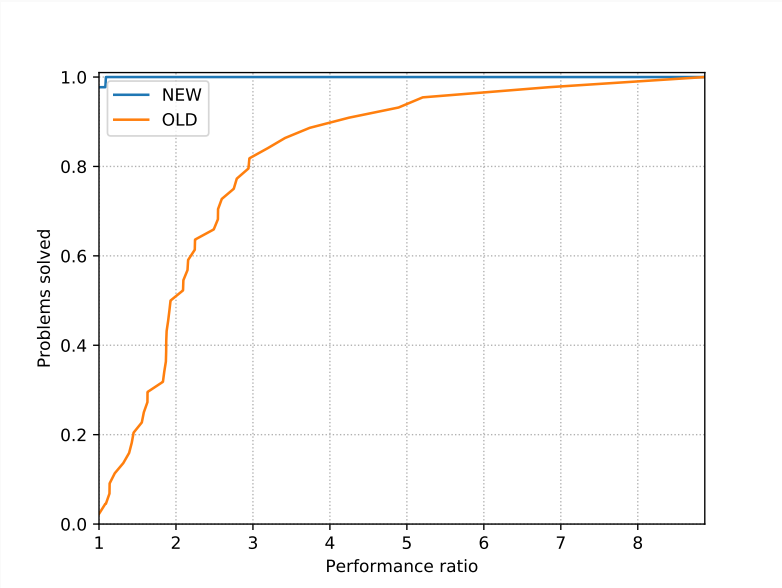
- Performance in Julia were first much worsed than in Ampl/Matlab
- Instability: HSL MA27 and HSL MA57 give different results...

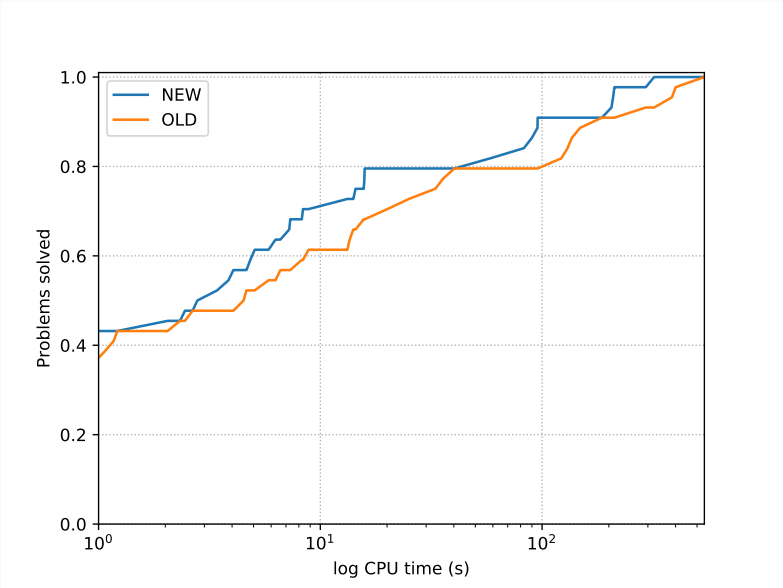
Results

We compare

OLD: Knitro.jl + MathProgBase

NEW: Knitro.jl + MathOptInterface





Code available in https://github.com/frapac/Knitro-Zoo/opf/

Extension

Some ideas to move forward...

- Challenge: implementing Security constraints OPF (MINLP & complementarity constraints) inside a Benders decomposition
- Add supports of NLP constraints/objective in StructJuMP.jl

Going onward

Contributing to JuMP?

- Callbacks?
- Implement Complementarity Constraints in MOI/JuMP?
- Implement non-linear least-squares in MOI/JuMP?
- NLP: first class support planned for JuMP 2.0
  Other promising projects:
  - Optim.jl
  - JuliaSmoothOptimizers.jl

State of AD in 2019

- Currently JuMP still relies on a specific version of ForwardDiff.jl
- Recent buzz around AD:
  https://julialang.org/blog/2019/01/fluxdiffeq
- Lots of development is ongoing:
  - Cassette.jl
  - Nabla.jl
  - Zygote.jl

Still looking for difficult (MI)NLP problems!

- Lots of benchmarks in GAMS/AMPL formats
- We believe the community will benefit from established benchmark built on top of JuMP
- Some very promising ideas related to MathOptFormat.jl

Wrapping it alltogethers

- Implementing a wrapper for a solver in Julia allows to understand lot of stuffs
- Next tutorial: solving MINLP problems with Knitro

- Artelys is recruiting (internship & CDI)!

francois.pacaud@artelys.com

https://www.artelys.com/fr/carrieres